

Introduction to Real-Time Operating Systems

GPOS vs RTOS

- General purpose operating systems
- Real-time operating systems

GPOS vs RTOS: Similarities

- Multitasking
- Resource management
- OS services to applications
- Abstracting the hardware

Characteristics of RTOS

- Reliability in embedded application
- Scale up or down ability
- Faster performance
- Reduced memory requirement
- Scheduling policies for real-time
- Diskless
- portability

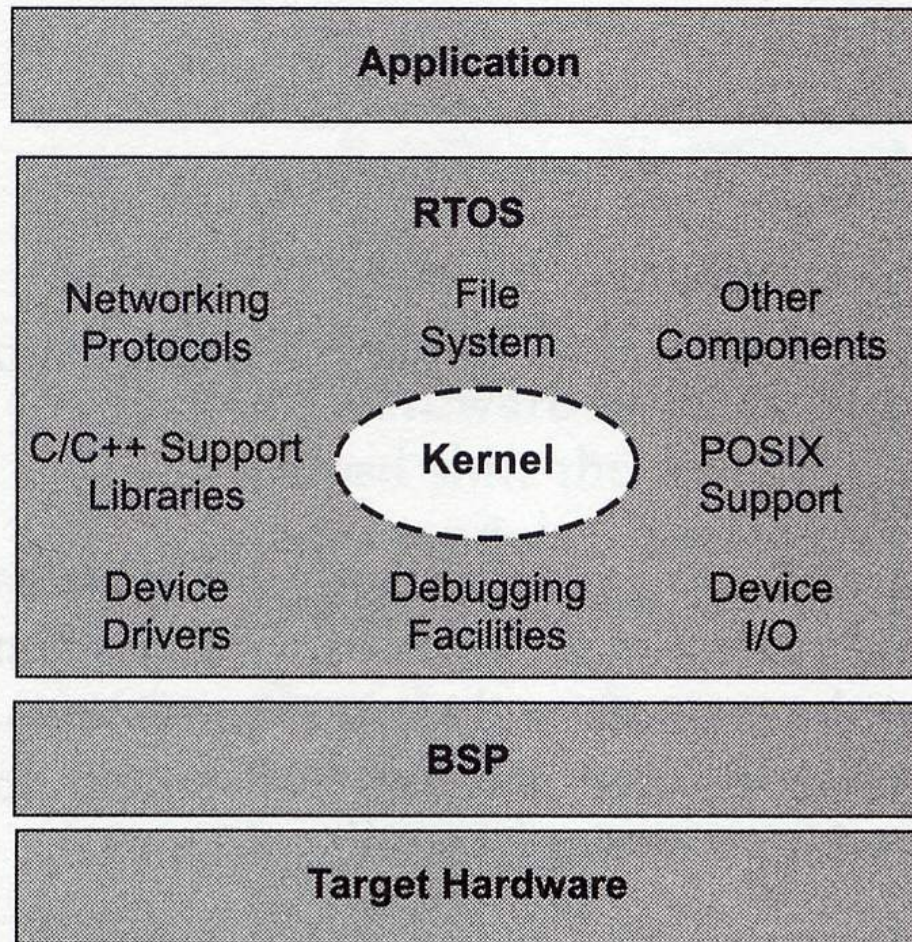


Figure 4.1 High-level view of an RTOS, its kernel, and other components found in embedded systems.

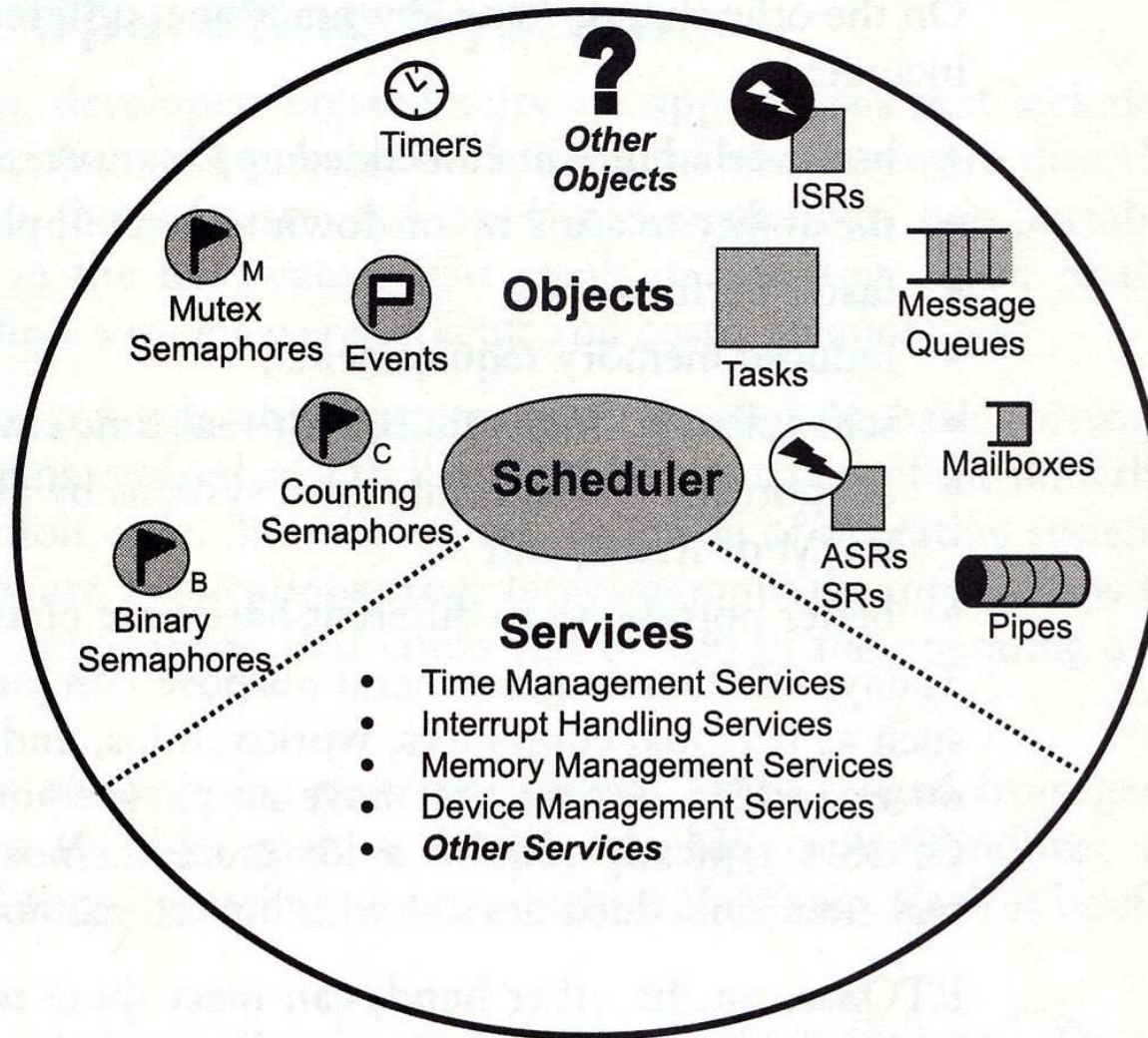


Figure 4.2 Common components in an RTOS kernel that including objects, the scheduler, and some services.

Kernel Objects

- Help developers creates applications for real-time embedded systems

Scheduler

- Determine which task executes when
- Schedulable entities-a kernel object that can compete for execution on a system-> process, task
- Multitasking: many thread of execution appear to be running concurrently

Scheduler

- Context: the state of CPU registers
- Context switch
- When a new task is created, TCB(task control block) is also created
- TCB: system data structure

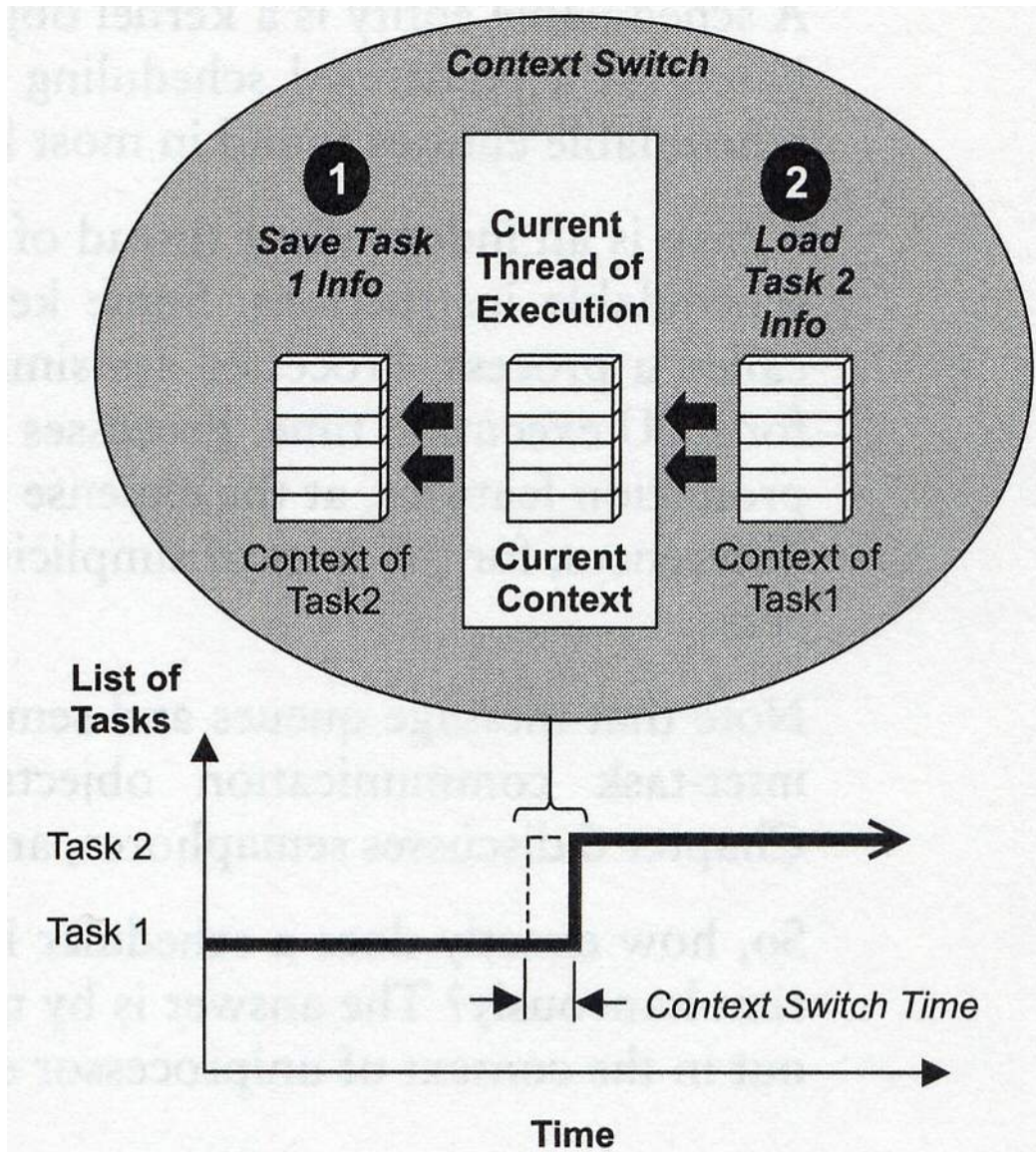


Figure 4.3 Multitasking using a context switch.

Scheduling Algorithms

- Preemptive priority-based scheduling
- Round-robin scheduling

- 256 priority levels
- 0: highest
- 256: lowest

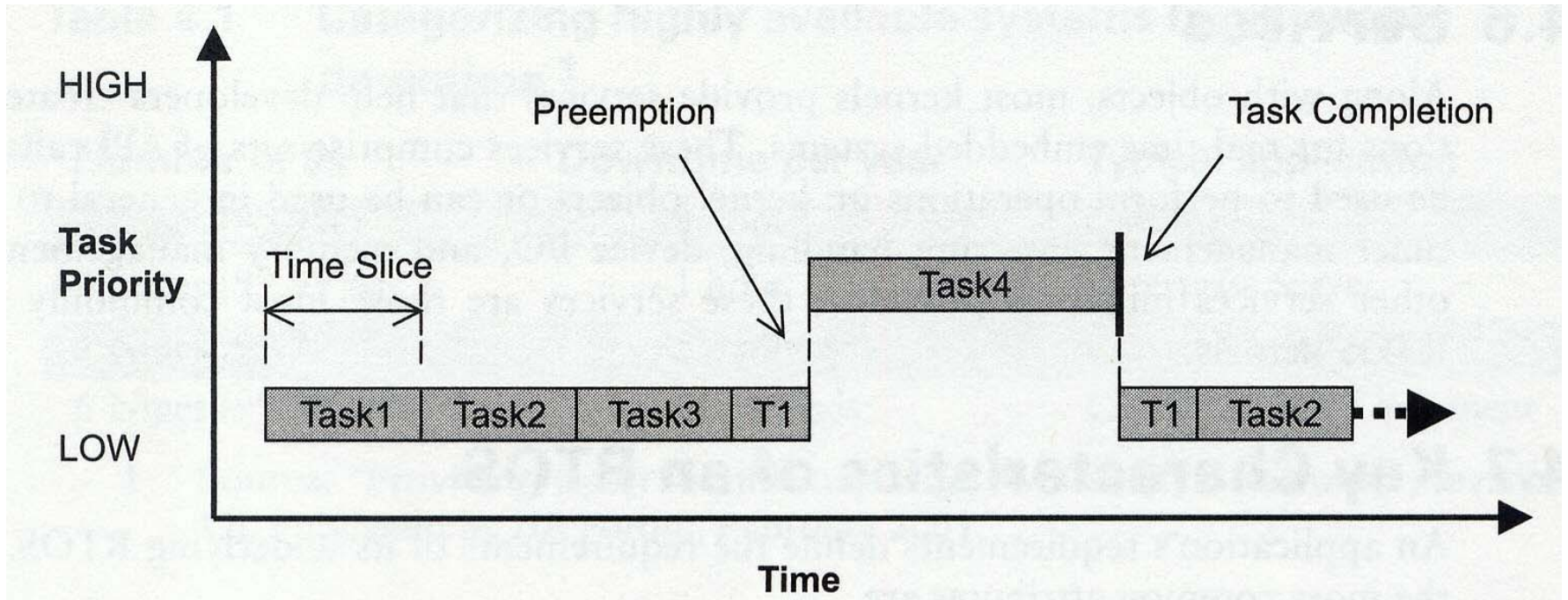


Figure 4.5 Round-robin and preemptive scheduling.

Objects

- Tasks
- Semaphore: token-like objects for synchronization & mutual exclusion
- Message queue: buffer-like data structures

Common Real-Time Design Problems

- Concurrency
 - Activity synchronization
 - Data communication
-
- Developers combine basic kernel objects

Key Characteristics of RTOS

- Reliability
- Predictability
- Performance
- Compactness
- Scalability