Embedded Motion Control 4K450

Lecture 2 – Chapter 4:

Introduction to real-time operating systems

Group 2

Technische Universiteit **Eindhoven** University of Technology

Where innovation starts

Introduction

Real-time operating systems (RTOS)

- A brief history of operating systems
- A definition of a RTOS
- Scheduler
- Objects
- Services
- Key characteristics
- Summary

History of Operating Systems

- Early days: interact with hardware directly
- Later: abstraction of underlying hardware
  - General-Purpose OS:
    - UNIX
    - Windows
  - Real-Time OS
    - VxWorks

Similarities GPOS & RTOS

- Multitasking
- Resource Management
- Provision of OS services to applications
- Abstract hardware from software

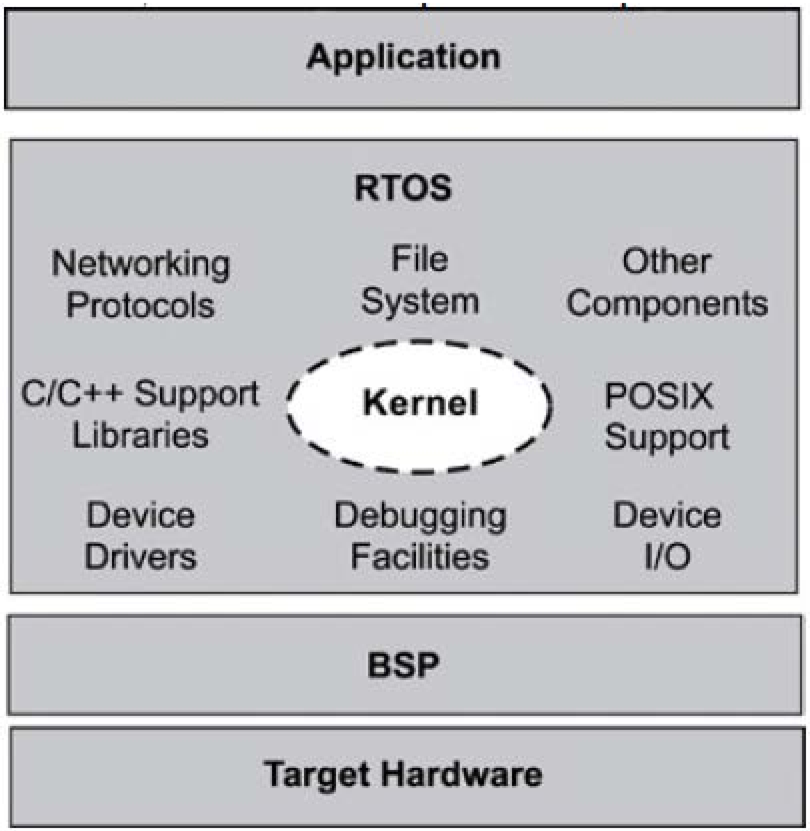
Differences RTOS

- Better reliability
- Ability to scale up or down
- Faster performance
- Reduced memory requirements
- Scheduling policies tailored for real-time embedded systems
- Better portability

Defining Real-Time OS (1/2)

- **Program that schedules execution in a timely manner, manages system resources, and provides a consistent foundation for developing application code**

Defining Real-Time OS (2/2)



Every RTOS:

- Kernel

Most RTOS kernels:

- Scheduler
- Objects
- Services

The Scheduler

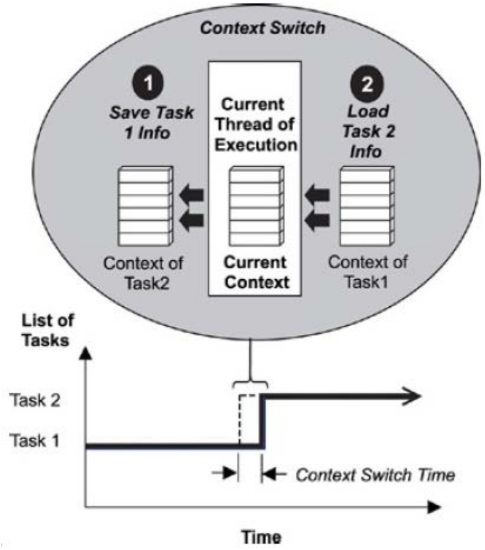
- Provides the algorithms to determine which task executes when.

- Scheduable entities
  - Kernel objects that compete for execution time
- Task or Process
  - Independent thread that contains independently scheduable instructions

Technische Universiteit
Eindhoven
University of Technology

Multitasking

- The ability to handle multiple activities within set deadlines
- Threads might seem to be running concurrently, however the kernel is interleaving them sequentially
- Based on scheduling algorythm
- More tasks = higher requirements



The Context Switch

- Task Control Block (TCB) is data structure for a task
- When the kernel switches from one task to another:
  - It saves task 1s context info in its TCB
  - It loads task 2s info from its TCB, this becomes current
  - Context of task 1 is frozen until scheduler switches to task 1 again
- Context switch time
  - Time it takes for scheduler to switch between tasks
  - Relatively insignificant, but excessive switching incurs unnecessary performance overhead

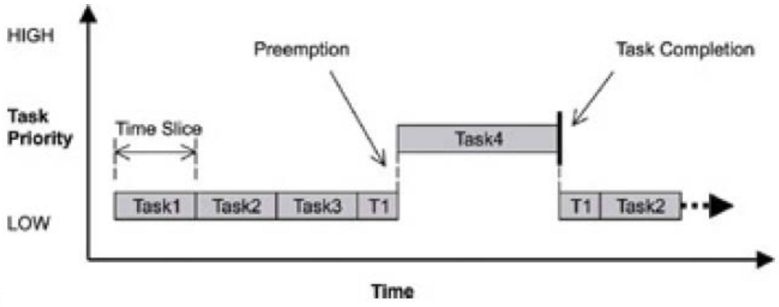
The Dispatcher

- Performs the context switching and changes the flow of execution
- Passes the flow of execution (control) through either an application task, through an ISR or through the kernel
- Which task should be executed is determined by the scheduling algorithm

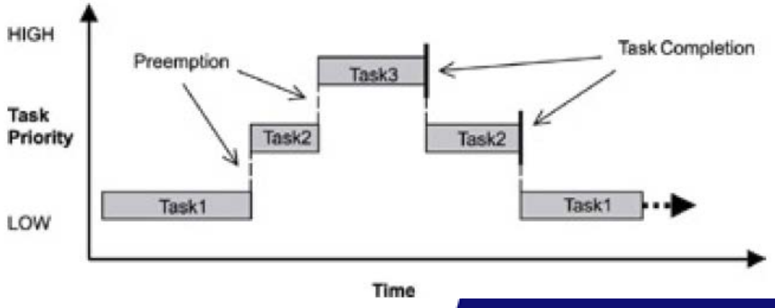
Scheduling algorithms

Scheduling algorithms support 256 priority levels

- Preemptive priority-based scheduling
  - Scheduling of tasks with different priority levels



- Round-robin scheduling
  - Scheduling of tasks with equal priority levels



Objects

Building blocks for application development

- **Taks**
  - **Concurrent and independent threads of execution**

- **Semaphores**
  - **Chapter 7**

- **Message Queues**
  - **Buffer-like data structures**

Services

- Application programming interface (API) calls
  - Bios

- Operations on kernel objects

- General:
  - Timer management
  - Interrupt handling
  - Device I/O
  - Memory management

Key Characteristics of an RTOS

- Reliability
- Predictability
- Performance
- Compactness
- Scalability

Key Characteristics of an RTOS

- **Reliability**
  - **Different degrees**
  - **Downtime per year, availability**
- Predictability
- Performance
- Compactness
- Scalability

Number of 9s	Downtime per year
3 Nines (99.9%)	~9 hours
4 Nines (99.99%)	~1 hour
5 Nines (99.999%)	~5 minutes
6 Nines (99.9999%)	~31 seconds

Key Characteristics of an RTOS

- Reliability
- **Predictability**
  - **Deterministic RTOS**
  - **Small variance of response times**
- Performance
- Compactness
- Scalability

Key Characteristics of an RTOS

- Reliability
- Predictability
- Performance
  - Meet time requirements
  - Million Instructions Per Second (MIPS)
  - Throughput: rate, bps
  - Call-by-call basis
- Compactness
- Scalability

Technische Universiteit
Eindhoven
University of Technology

Key Characteristics of an RTOS

- Reliability
- Predictability
- Performance
- Compactness
  - Application design constraints
  - Cost constraints
  - RTOS memory footprint
- Scalability

Key Characteristics of an RTOS

- Reliability
- Predictability
- Performance
- Compactness
- Scalability
  - Meet application-specific requirements
  - Save time and money

Summary

- RTOS vs GPOS
- RTOS definition
  - Schedule execution, manage system resources, provide foundation
- Kernels
  - Objects, services, scheduler
- Task scheduling
  - Preemptive, round-robin
- Key characteristics
  - Reliable, predictable, high performance, compact, scalable