SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECT312 – EMBEDDED SYSTEM DESIGN

III YEAR/ VI SEMESTER

1

TOPIC 3 & 4 : Introduction to Real-Time Operating Systems (RTOS)

What is an Operating System ?

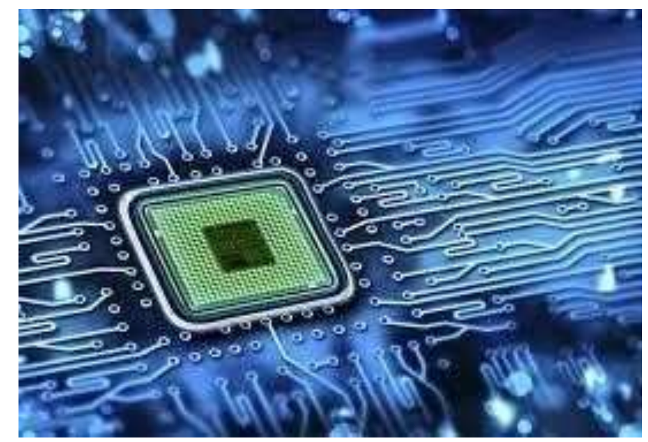
- An operating system (OS) is a fundamental software program that acts as an intermediary between computer hardware and user applications.
- It manages computer resources, such as memory, processing units, and peripheral devices, and provides services for executing programs and facilitating user interactions.
- Operating systems enable users to interact with computers efficiently by providing interfaces through which users can run applications, manage files, and control system settings.
- Additionally, operating systems incorporate essential functionalities like process management, memory management, file system management, device management, user interface management, security enforcement, and networking capabilities.

Real-Time Operating Systems (RTOS)

Definition of RTOS

- A Real-Time Operating System (RTOS) is a specialized operating system designed to handle tasks with specific timing constraints and stringent requirements for predictability and responsiveness.

- Unlike general-purpose operating systems, which prioritize tasks based on factors such as fairness and efficiency, RTOSes prioritize tasks based on their deadlines and importance to meet real-time requirements.



Real-Time Operating Systems (RTOS)

key features :

Task Scheduling : An RTOS schedules tasks based on their priority, ensuring that critical tasks are executed within their specified time constraints.

Interrupt Handling: An RTOS efficiently handles interrupts, allowing for quick response times to time-sensitive events.
Resource Management: An RTOS manages system resources, such as memory and peripherals, to ensure efficient and reliable operation.

Communication and Synchronization: An RTOS provides mechanisms for inter-task communication and synchronization, enabling tasks to exchange data and coordinate their activities.

Real-Time Operating Systems (RTOS)

Why Use an RTOS?

Precise Timing

Real-time operating systems (RTOS) provide precise timing capabilities, allowing for accurate control and synchronization of tasks and events.

Task Prioritization

An RTOS allows for task prioritization, ensuring that critical tasks are executed in a timely manner and higher-priority tasks are given precedence.

Resource Management

RTOSs provide efficient resource management, allowing for optimal utilization of system resources such as CPU, memory, and peripherals.

Simplified Application Development

Using an RTOS can simplify application development by providing a framework for task management, inter-task communication, and synchronization.

Improved System Reliability

By providing deterministic behavior and isolation of tasks, RTOSs can improve system reliability and robustness.

Real-Time Operating Systems (RTOS)

Task Scheduling in an RTOS

- In a Real-Time Operating System (RTOS), task scheduling is a critical aspect of ensuring that tasks are executed in a timely manner.
- The goal of task scheduling is to assign priorities to tasks and determine the order in which they are executed. This ensures that critical tasks are given higher priority and are executed within their deadlines.

Task scheduling in an RTOS involves the following steps:

Task Prioritization: Each task is assigned a priority level based on its importance and urgency. Critical tasks are assigned higher priority levels, while less critical tasks are assigned lower priority levels.

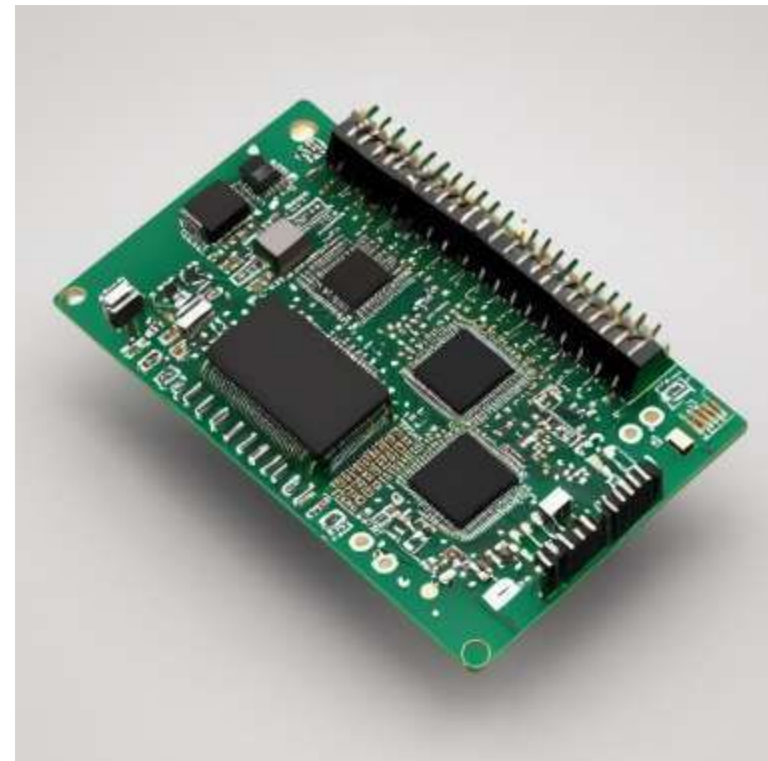
Task Execution Order: The RTOS determines the order in which tasks are executed based on their priority levels. Higher priority tasks are executed before lower priority tasks.

Task Preemption: If a higher priority task becomes ready to run while a lower priority task is currently executing, the RTOS may preempt the lower priority task and switch to executing the higher priority task.

Task Scheduling Algorithm: The RTOS uses a task scheduling algorithm, such as a priority-based or round-robin algorithm, to determine the execution order of tasks.

Interrupt Handling in an RTOS

- Interrupt handling in an RTOS involves managing and responding to interrupts from external devices.
- This ensures that critical events are handled in a timely manner without disrupting the execution of other tasks.



Memory Management in an RTOS

Allocating and Deallocating Memory

- In an RTOS, memory management involves allocating and deallocating memory resources for tasks.
- This ensures that each task has sufficient memory to execute its operations efficiently.

Efficient Memory Usage

- Efficient memory usage is crucial in an RTOS to prevent memory leaks and improve system stability.
- By managing memory effectively, the RTOS can optimize the allocation and deallocation of memory resources, reducing the risk of memory-related errors.

Real-Time Operating Systems (RTOS)

RTOS Examples

FreeRTOS
Features: Open-source, small footprint, real-time kernel
Use Cases: Embedded systems, IoT devices, consumer electronics

VxWorks
Features: Real-time, deterministic performance, high reliability
Use Cases: Aerospace, defense, industrial automation

QNX
Features: Real-time, microkernel architecture, high availability
Use Cases: Automotive, medical devices, telecommunications

Real-Time Operating Systems (RTOS)

RTOS Limitations

Increased Complexity

RTOS introduces additional complexity to the system design and development process. Real-time scheduling, task synchronization, and resource management require careful consideration and implementation.

Higher Resource Requirements

RTOS typically requires more system resources compared to non-real-time operating systems. Real-time tasks and scheduling algorithms consume additional memory and processing power.

Potential for Priority Inversion

Priority inversion occurs when a low-priority task holds a resource required by a higher-priority task, leading to delays and potential system failures. Proper priority assignment and synchronization mechanisms are necessary to prevent priority inversion.

Real-Time Operating Systems (RTOS)

RTOS Applications

RTOS has various applications in industries such as:

- Aerospace
- Automotive
- Medical
- Industrial Automation

RTOS is used in systems that require precise timing and reliable operation

Real-Time Operating Systems (RTOS)

Conclusion :

- In conclusion, Real-Time Operating Systems (RTOS) are essential for ensuring timely and predictable responses in critical applications.
- From medical devices to automotive systems, RTOS plays a vital role in maintaining safety, reliability, and performance.
- As technology advances, the importance of RTOS will only continue to grow, driving innovation and enhancing real-time computing capabilities across industries

SUMMARY & THANK YOU