COLLEGE OF ENGINEERING & MANAGEMENT KOLAGHAT

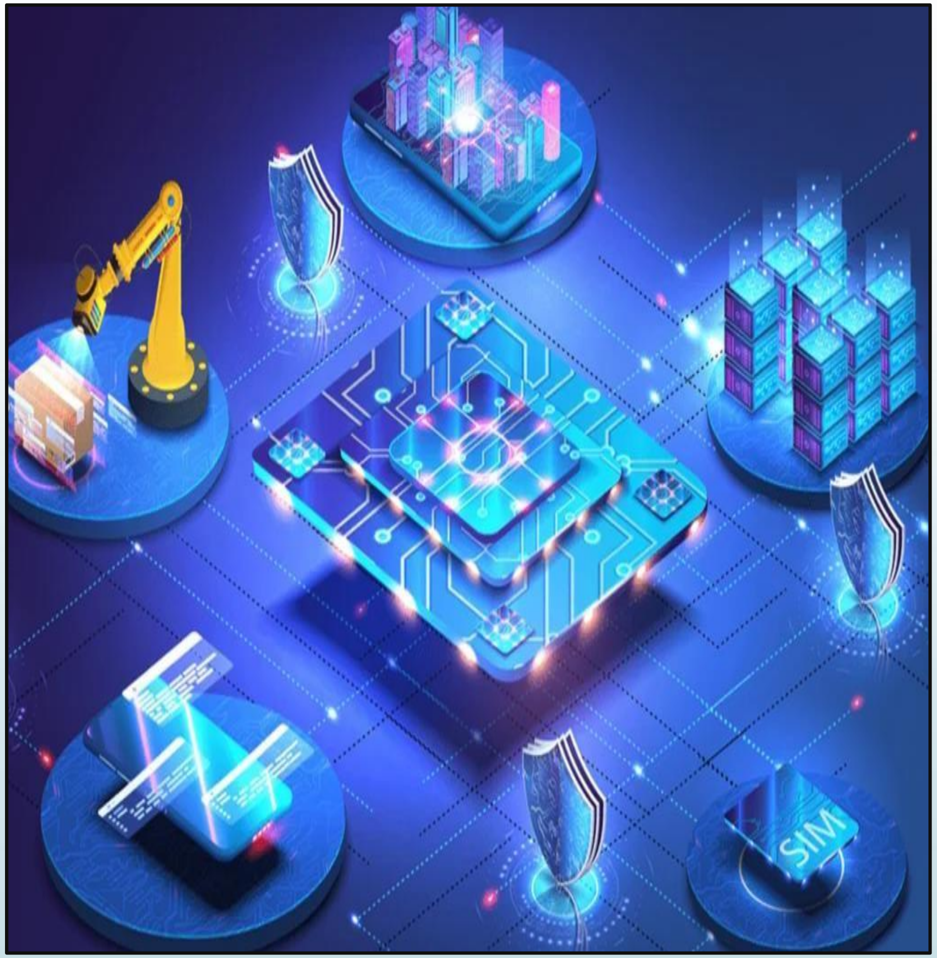
SUBJECT - OPERATING SYSTEM.
SUBJECT CODE – PCC - CS502
TOPIC –REAL TIME OPERATING SYSTEM.

NAME – SANJAY MAITY
UNIVERSITY ROLL NO – 10700123198
UNIVERSITY REG. NO – 231070120412
COLLEGE ROLL NO – CSE/23/L193
DEPARTMENT – COMPUTER SCIENCE AND ENGINEERING
SECTION – C3
ACADEMIC SESSION – 2024 - 2025

TABLE OF CONTENT

REAL-TIME OPERATING SYSTEMS: AN INTRODUCTION

- Real-time operating systems (RTOS) are designed for applications where the timely completion of tasks is critical. They are characterized by their ability to respond to events in a predictable and timely manner, often within strict deadlines. RTOS are prevalent in industries where time-critical operations are essential, such as aerospace, automotive, industrial automation, and medical devices. These systems are fundamentally different from general-purpose operating systems like Windows or macOS, which are designed for desktop and server applications where responsiveness is important but not time-critical.



BACKGROUND STUDY

The development of RTOS traces back to the early days of computing, with the emergence of industrial automation, aerospace, and military applications.

- 1 **Early Automation (1950s)**
  - Initial applications of RTOS emerged in industrial automation and control systems, where timely responses were crucial.
- **2 Space Exploration (1960s)**
  - RTOS became vital in space exploration, enabling spacecraft guidance, navigation, and control systems to operate in realtime.
- **3 Modern Era (1980s-Present)**
  - RTOS have expanded into diverse applications, including mobile devices, embedded systems, and the Internet of Things (IoT).

CHARACTERISTICS OF REAL-TIME SYSTEMS

Deterministic Behaviour:

- RTOS are designed to provide predictable and consistent response times. They strive to minimize the variability in execution time for tasks, ensuring that operations happen within a predictable timeframe. This deterministic behaviour is crucial for applications where timing errors can lead to safety hazards or functional failures.

Resource Management:

- RTOS meticulously manage system resources like memory, CPU time, and peripherals to ensure timely task execution. They employ mechanisms to avoid resource contention and ensure that critical tasks are allocated the necessary resources without delay. This is essential for maintaining the realtime performance of the system.

Event-Driven Architecture

- RTOS are built on an event-driven architecture, where they respond to events like sensor readings, user inputs, and network messages. These events trigger the execution of specific tasks or interrupt handlers, ensuring a rapid and efficient response to real-time demands.

THEORY DISCUSSION:

- RTOS theory revolves around key concepts like task scheduling, interrupt handling, and resource management, ensuring predictable and timely responses to events

TASK SCHEDULING IN REAL-TIME OS:

1.Priority-Based Scheduling:

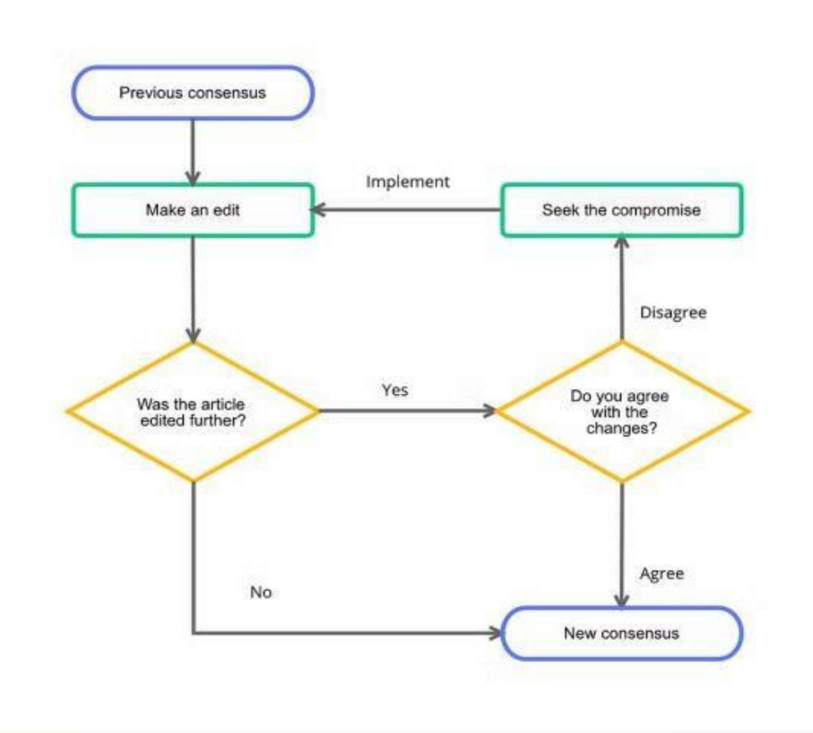
- This common approach assigns priorities to tasks based on their criticality. High-priority tasks are given preferential access to the CPU, ensuring they are executed promptly. This helps guarantee that time-critical operations are handled without delay.

2 Rate Monotonic Scheduling:

- A deterministic scheduling algorithm that assigns priorities based on the frequency of tasks. Higher-frequency tasks are given higher priority, ensuring that they are executed regularly within their deadlines. This method is widely used in systems where real-time performance is paramount.

3 Earliest Deadline First (EDF):

- EDF assigns priorities based on the deadlines of tasks. Tasks with earlier deadlines are given higher priority. This ensures that tasks with the most imminent deadlines are completed first, minimizing the chance of missed deadlines.



INTERRUPT HANDLING IN REALTIME OS

1 Fast Interrupt Handling

- RTOS prioritize fast interrupt handling to ensure timely responses to external events. Interrupts are used to signal events like sensor readings or user inputs, and the OS must respond quickly to these events to maintain real-time performance.

2 Interrupt Latency Minimization

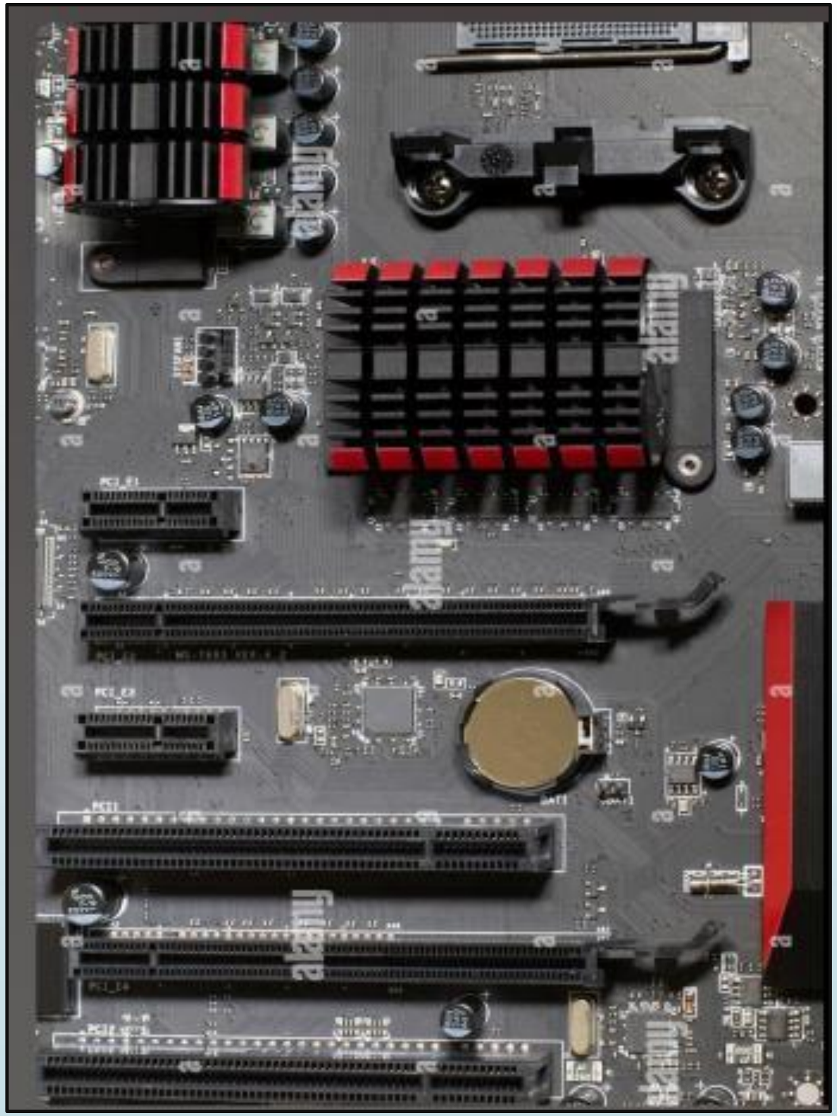
- Minimizing interrupt latency is crucial for real-time systems. Latency is the delay between an interrupt occurring and the system responding to it. RTOS employ techniques like interrupt prioritization and interrupt coalescing to reduce latency and ensure responsiveness.

3 Interrupt Disabling

- RTOS can temporarily disable interrupts during critical operations to avoid unexpected interruptions. This helps ensure that tasks requiring uninterrupted access to resources can complete without delays caused by external events.

4 Interrupt Context Switching

- When an interrupt occurs, the RTOS switches the CPU context from the currently running task to the interrupt handler. This ensures that the interrupt is handled promptly and efficiently, without interrupting the execution of critical tasks.



MEMORY MANAGEMENT IN REAL-TIME OS

1.Fixed-Priority Memory Allocation

- A simple method where memory is allocated to tasks based on their priority. High-priority tasks are given priority access to memory, ensuring they have the resources they need to execute promptly.

2 Real-time Garbage Collection

- RTOS can use real-time garbage collection algorithms to reclaim unused memory efficiently. These algorithms prioritize the timely release of memory, ensuring that it is available for critical tasks without delays.

3 Memory Protection

- Memory protection is crucial in real-time systems to prevent tasks from interfering with each other's memory spaces. This helps ensure the integrity of data and prevent errors that could compromise the system's real-time performance.

TYPES OF REAL TIME OPERATING SYSTEM

1. **Hard Real-Time Operating System**

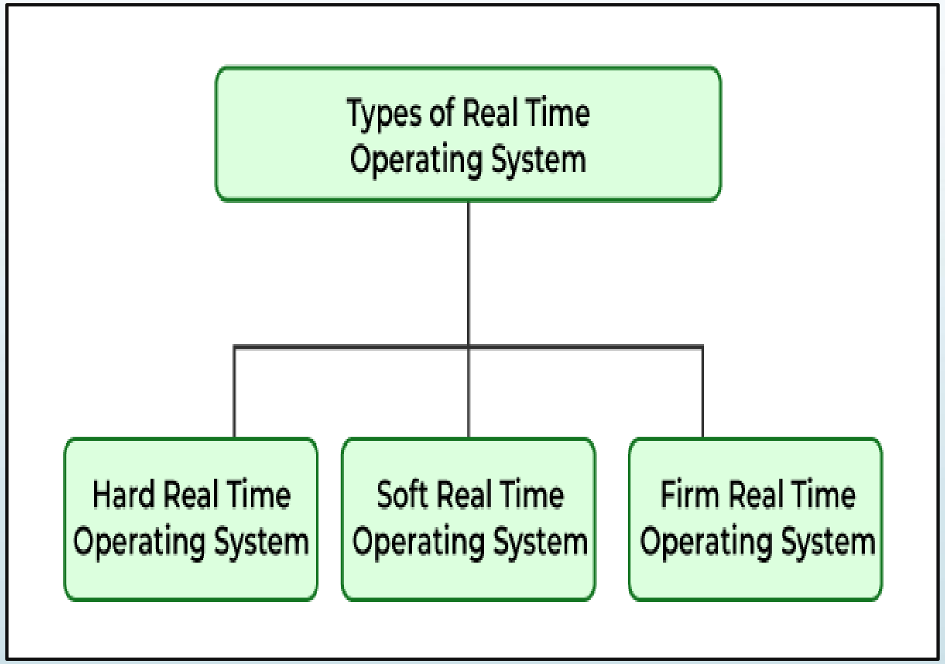
   - These operating systems guarantee that critical tasks are completed within a range of time. For example, a robot is hired to weld a car body. If the robot welds too early or too late, the car cannot be sold, so it is a hard real-time system that requires complete car welding by the robot hardly on time., scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

2. **Soft Real-Time Operating System**

   - This operating system provides some relaxation in the time limit. For example – Multimedia systems, digital audio systems, etc. Explicit, programmer-defined, and controlled processes are encountered in real-time systems. A separate process is changed by handling a single external event. The process is activated upon the occurrence of the related event signaled by an interrupt..

3. **Firm Real-time Operating System**

   - RTOS of this type have to follow deadlines as well. In spite of its small impact, missing a deadline can have unintended consequences, including a reduction in the quality of the product. Example: Multimedia applications.



ADVANTAGES OF REAL-TIME OPERATING SYSTEMS

- RTOS offer distinct advantages that make them well-suited for time-critical applications.

1. **Timely Responses:**
   - RTOS guarantee timely responses to external events, crucial for systems with strict deadlines.
2. **Predictable Performance**
   - RTOS provide deterministic behaviour, minimizing latency and variability in response times.
3. **Resource Efficiency**
   - RTOS efficiently manage system resources, ensuring optimal utilization for real-time operations.
4. **Enhanced Reliability**
   - RTOS prioritize system stability and reliability, crucial for safety-critical applications.

APPLICATIONS OF REAL-TIME OPERATING SYSTEMS

RTOS have found widespread applications across various industries and technologies

Aerospace

RTOS are critical in aerospace applications, controlling aircraft systems, navigation, and flight control. They provide the reliability and real-time performance required for safe and efficient aircraft operation.

Automotive

RTOS are widely used in modern vehicles, controlling engine management, braking systems, infotainment systems, and driver assistance features. They provide the real-time performance required for safe and efficient vehicle operation.

Industrial Automation

RTOS are essential in industrial automation, controlling robots, manufacturing processes, and other industrial equipment. They provide the reliability and real-time performance required for efficient and reliable industrial operations

FUTURE TRENDS IN REAL-TIME OPERATING SYSTEMS

The future of real-time operating systems is shaped by evolving technologies and emerging applications. Some key trends include:

- Increased integration with artificial intelligence (AI) and machine learning (ML) for improved decision-making and adaptive behaviour in real-time systems.
- Advancements in cloud-based RTOS for greater scalability, flexibility, and remote management of real-time applications.
- Growing use of real-time operating systems in edge computing environments for data processing and decisionmaking at the edge of the network.
- Development of real-time operating systems for quantum computing platforms, opening up new possibilities for high-performance computing and real-time applications.

CONCLUSION

- A Real-Time Operating System (RTOS) is designed to manage hardware resources and run applications in a way that meets specific timing requirements. Unlike general-purpose operating systems, RTOS focuses on deterministic behavior and timely task completion, making it crucial for applications where timing is critical, such as embedded systems, industrial automation, and medical devices.
- The key advantages of an RTOS include predictable response times, high reliability, and efficient resource management. It prioritizes tasks to ensure that high-priority processes are executed within the required time constraints. This predictability is achieved through features like preemptive multitasking, real-time scheduling algorithms, and interrupt handling.
- However, RTOS also comes with challenges, such as increased complexity in system design and a steeper learning curve for developers. Additionally, the need for precise timing can limit the flexibility and general-purpose capabilities of the system.
- In conclusion, RTOS is essential for applications where timing and reliability are paramount. While it introduces complexity, its ability to provide deterministic and predictable responses makes it indispensable in fields requiring precise and timely operations.

REFERENCES

- https://www.geeksforgeeks.org/real-time-operating-system-rtos/
- https://en.wikipedia.org/wiki/Real-time_operating_system
- https://www.javatpoint.com/real-time-operating-system
- https://www.toppr.com/guides/computer-science/computer-fundamentals/operating-system/real-time-operating-system-rtos/
- https://www.techtarget.com/searchdatacenter/definition/real-time-operating-system
- https://www.spiceworks.com/tech/hardware/articles/what-is-rtos/
- https://www.ni.com/en/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-labview-real-time-module/what-is-a-real-time-operating-system--rtos--.html

Thank You