Introduction to Real-Time Systems

---

1. Definition of Real-Time Systems

A **Real-Time System (RTS)** is a computing system that must respond to inputs or events **within a specified time constraint**. The **correctness** of a real-time system depends **not only on the logical result** of the computation but also on the **time at which the result is produced**.

🔑 Key Characteristics:

- **Deterministic**: Behavior is predictable under defined conditions.
- **Deadline-driven**: Tasks must be completed within fixed time frames.
- **Concurrency**: Ability to handle multiple tasks simultaneously.
- **Reliability**: High availability and fault tolerance.
- **Embedded nature**: Often embedded within hardware systems.

---

2. Types of Real-Time Systems

1. Hard Real-Time Systems

- **Definition**: Missing a deadline is considered a total system failure.
- **Examples**:
  - Aircraft control systems
  - Pacemakers
  - Nuclear reactor monitoring
- **Characteristics**:
  - Guaranteed response times
  - Strict scheduling and timing constraints

2. Soft Real-Time Systems

- **Definition**: Occasional deadline misses are tolerated, but performance degrades.
- **Examples**:
  - Multimedia streaming
  - Online transaction systems
  - Video conferencing
- **Characteristics**:
  - Focus on throughput and responsiveness
  - Tolerates delay but aims for minimal latency

3. Firm Real-Time Systems

- **Definition**: Missing deadlines leads to useless output, but not catastrophic failure.
- **Examples**:
  - Stock trading systems
  - Weather data collection
- **Characteristics**:
  - Deadlines are important but not strictly enforced
  - Output becomes irrelevant if late

---

3. Real-Time Applications in Various Domains

Domain	Example Applications
Automotive	Engine control units, adaptive cruise control
Aerospace	Flight control systems, space probes
Industrial Control	Robotic arms, CNC machines
Healthcare	Medical monitoring systems, infusion pumps
Telecommunications	Real-time call processing, video streaming servers
Finance	High-frequency trading systems, fraud detection
Military	Radar systems, missile guidance
Consumer Electronics	Gaming consoles, smart TVs, real-time home automation

---

4. Real-Time Operating System (RTOS) Concepts

An **RTOS** is a specialized operating system designed to handle real-time tasks.

🎯 Key Functions:

- **Task Scheduling**: Ensures tasks meet deadlines.
- **Interrupt Handling**: Quickly responds to events.
- **Inter-process Communication (IPC)**: Manages data exchange between tasks.
- **Priority Management**: Assigns and handles task priorities.

🛠️ RTOS Scheduling Algorithms:

- **Rate Monotonic Scheduling (RMS)**: Fixed priority; shorter periods = higher priority.
- **Earliest Deadline First (EDF)**: Dynamic priority; task with nearest deadline runs first.
- **Round Robin** (in soft real-time systems): Equal CPU time slice.

🔁 Key RTOS Components:

- **Scheduler**: Determines which task runs next.
- **Clock & Timers**: Keeps track of system time.
- **Memory Manager**: Allocates memory deterministically.
- **I/O Subsystem**: Ensures timely interaction with external devices.

🎁 Examples of RTOS:

- **FreeRTOS**
- **VxWorks**
- **RTLinux**
- **QNX**
- **Zephyr**

Summary

- Real-time systems require timely, deterministic responses.
- They are categorized as hard, soft, or firm based on deadline criticality.
- RTOS plays a vital role in managing real-time tasks efficiently.
- Applications span aerospace, automotive, healthcare, and many other sectors

Multiple Choice Questions (MCQs)

1. **Which of the following best defines a real-time system?**
   A. A system that always provides the correct output
   B. A system that completes tasks within a specific deadline
   C. A system that never fails
   D. A system that uses a graphical user interface
2. **In a hard real-time system, missing a deadline results in:**
   A. A slight delay
   B. Degraded performance
   C. System failure
   D. Rescheduling the task
3. **Which of the following is a soft real-time application?**
   A. Pacemaker
   B. Missile guidance
   C. Video streaming
   D. Nuclear reactor control
4. **What is a key feature of firm real-time systems?**
   A. Every deadline must be met
   B. Missing a deadline causes data loss, not failure

C. Deadline misses are tolerated always
D. Tasks are never scheduled dynamically

5. **Which of the following is NOT a characteristic of real-time systems?**
   A. Determinism
   B. Deadline-driven execution
   C. High-latency tolerance
   D. Concurrency
6. **An example of a hard real-time system is:**
   A. Online shopping website
   B. Heartbeat monitoring system
   C. Weather forecasting application
   D. Online gaming
7. **Real-time systems are commonly used in which of the following?**
   A. Word processors
   B. Data entry systems
   C. Industrial automation
   D. Email clients
8. **What is the main function of a Real-Time Operating System (RTOS)?**
   A. Speed up the boot time
   B. Manage hardware drivers
   C. Schedule tasks with time constraints
   D. Provide a graphical interface
9. **Which RTOS scheduling algorithm gives the highest priority to tasks with the shortest period?**
   A. Round Robin
   B. Rate Monotonic Scheduling
   C. First Come First Serve
   D. Shortest Job First
10. **Which algorithm assigns dynamic priority based on deadline proximity?**
    A. Rate Monotonic
    B. Earliest Deadline First
    C. Fixed Priority
    D. Time Slice Scheduling
11. **The ability of a real-time system to handle multiple tasks at once is called:**
    A. Interfacing
    B. Scheduling
    C. Concurrency
    D. Polling
12. **What does RTOS use to switch between tasks?**
    A. ROM
    B. Context switching
    C. Paging
    D. Memory swapping
13. **Which of the following is a real-time operating system?**
    A. Ubuntu
    B. Windows 11

C. FreeRTOS
D. Fedora

14. **Which of the following real-time applications can tolerate minor delays?**
    A. Flight control system
    B. Pacemaker
    C. Online video streaming
    D. Industrial robot arm
15. **What happens when a deadline is missed in a firm real-time system?**
    A. Task is retried later
    B. Task is canceled and result discarded
    C. System restarts
    D. Task is queued for next cycle
16. **Which of the following is NOT an example of a real-time system?**
    A. ATM withdrawal system
    B. Microwave oven controller
    C. Image editing software
    D. Traffic signal controller
17. **Which real-time OS component determines which task runs next?**
    A. Timer
    B. Memory manager
    C. Scheduler
    D. I/O handler
18. **Which of the following terms refers to time-bound execution of tasks?**
    A. Latency
    B. Throughput
    C. Determinism
    D. Deadline
19. **Which industry heavily relies on real-time systems for safety?**
    A. Fashion
    B. Agriculture
    C. Aerospace
    D. Social Media
20. **Which component in RTOS helps in inter-task data exchange?**
    A. Compiler
    B. Scheduler
    C. IPC (Inter-Process Communication)
    D. Timer

Practical Session: Exploring Real-Time Systems and RTOS Concepts

🎯 Objectives:

- Understand the differences between general-purpose OS and RTOS behavior.
- Demonstrate task scheduling and deadline management.
- Simulate hard and soft real-time systems.
- Get hands-on experience with a basic RTOS like **FreeRTOS** on a PC or microcontroller (or use an online simulator if hardware is not available).

🛠️ Tools & Requirements:

- **Option 1 (Simulation on PC):**
  - Platform: Windows/Linux
  - Tools: FreeRTOS Simulator or Keil RTX (Windows), or QEMU
  - IDE: VS Code with FreeRTOS extension / Eclipse
- **Option 2 (Microcontroller - Optional):**
  - Hardware: Arduino or STM32 with FreeRTOS
  - IDE: Arduino IDE / STM32CubeIDE

🧪 Lab Activities

Activity 1: Simulating Task Scheduling with FreeRTOS

Goal: Create two periodic tasks (Task A and Task B) and observe real-time task switching.

Steps:

1. Set up a FreeRTOS environment on PC or embedded board.
2. Create two tasks:
   - Task A: Print "Task A running" every 1000 ms
   - Task B: Print "Task B running" every 500 ms
3. Assign priorities: Task A (low), Task B (high)

Expected Result:

- Task B preempts Task A due to higher priority.
- Console shows Task B output more frequently.

Activity 2: Demonstrating Deadline Handling

Goal: Simulate deadline violation in a firm real-time system.

Steps:

1. Modify Task A to simulate a delay (e.g., `vTaskDelay(2000)`).
2. Set a deadline (e.g., 1000 ms).
3. Use a timer to detect if Task A misses its deadline.
4. If deadline missed, discard Task A's output.

Expected Result:

- If Task A doesn't complete in 1000 ms, its result is ignored and log shows "Task A deadline missed".

Activity 3: Exploring Soft vs. Hard Real-Time Scenarios

Goal: Show impact of missed deadlines in both soft and hard real-time systems.

Steps:

1. Simulate a video player (soft real-time): Use a loop printing "Frame X rendered".
    - Skip some frames intentionally.
    - Observe that playback continues.
2. Simulate a pacemaker logic (hard real-time): Output must occur every 500 ms.
    - Introduce intentional delay.
    - Observe system logs an error or "System failure" if delay occurs.

Expected Result:

- Soft system continues with degraded performance.
- Hard system halts or logs critical failure on deadline miss.

💼 Post-Lab Questions:

1. What happened when you increased the delay in Task A?
2. How does task priority affect scheduling in FreeRTOS?
3. How would you handle a missed deadline in a real-life medical device?
4. Can you think of a situation where a soft real-time system might become dangerous?

🎁 Deliverables:

- Code snippets for all tasks.
- Screenshots of terminal/log output.
- Answer sheet for post-lab questions.