Lecture 5

Embedded Systems

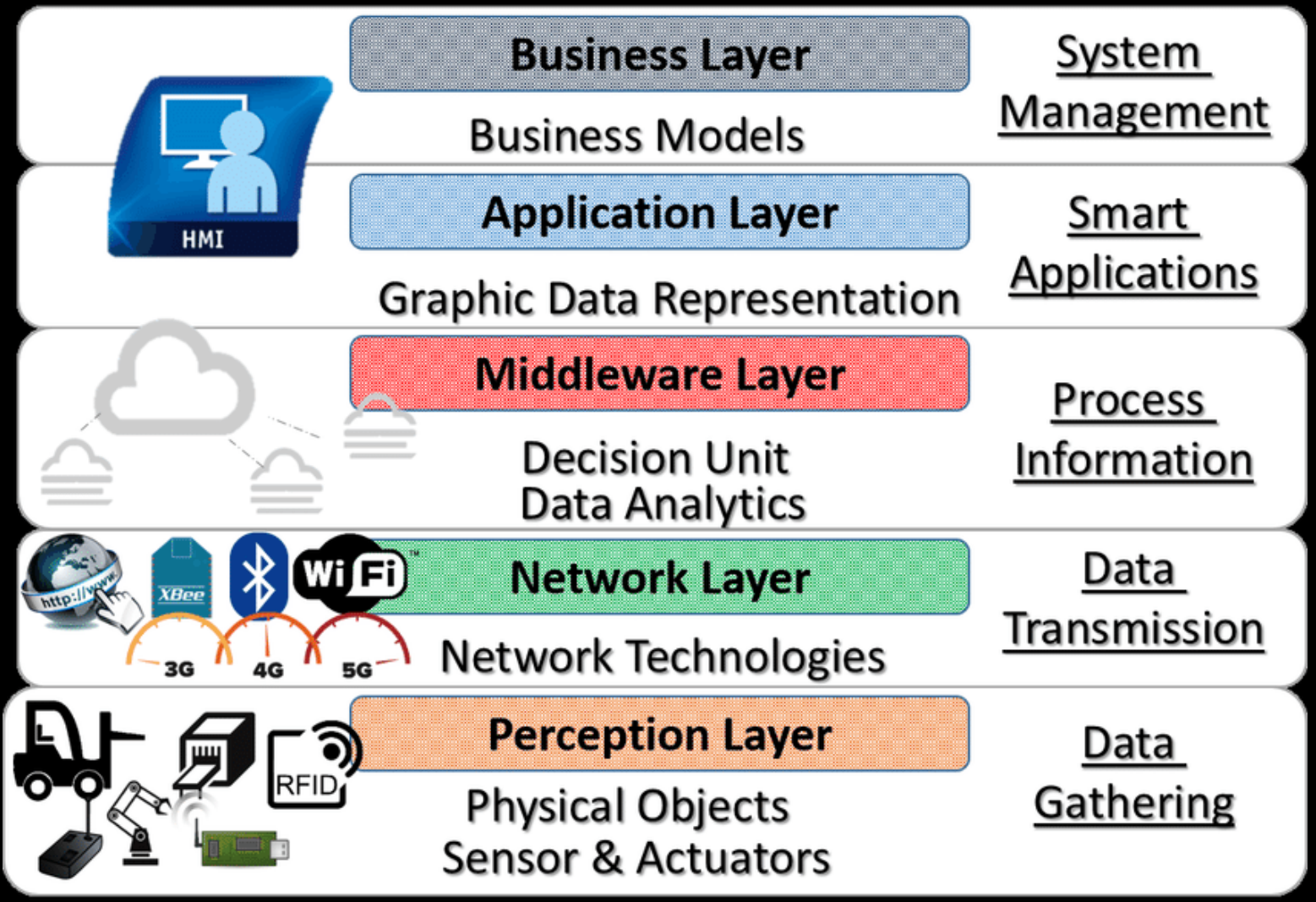
RTOS

Associated Prof. Wafaa Shalash

Lect. 5

Class Rules

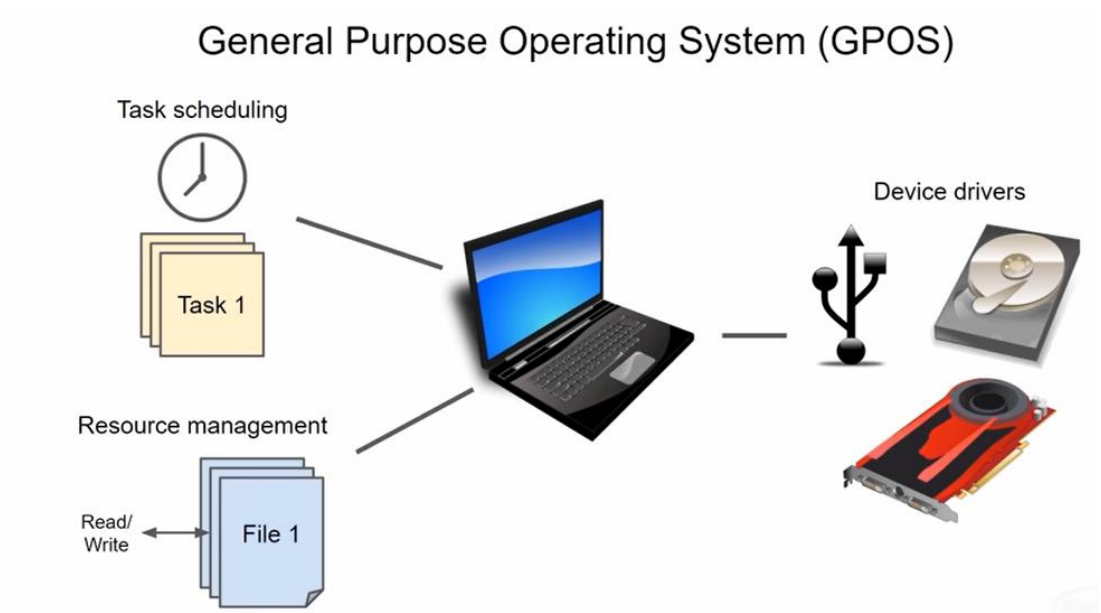
- Be in class on time,
- Listen to instructions and explanations.
- Talk to your classmates only when there is an activity.
- Use appropriate and professional language.
- Keep your mobile silent.

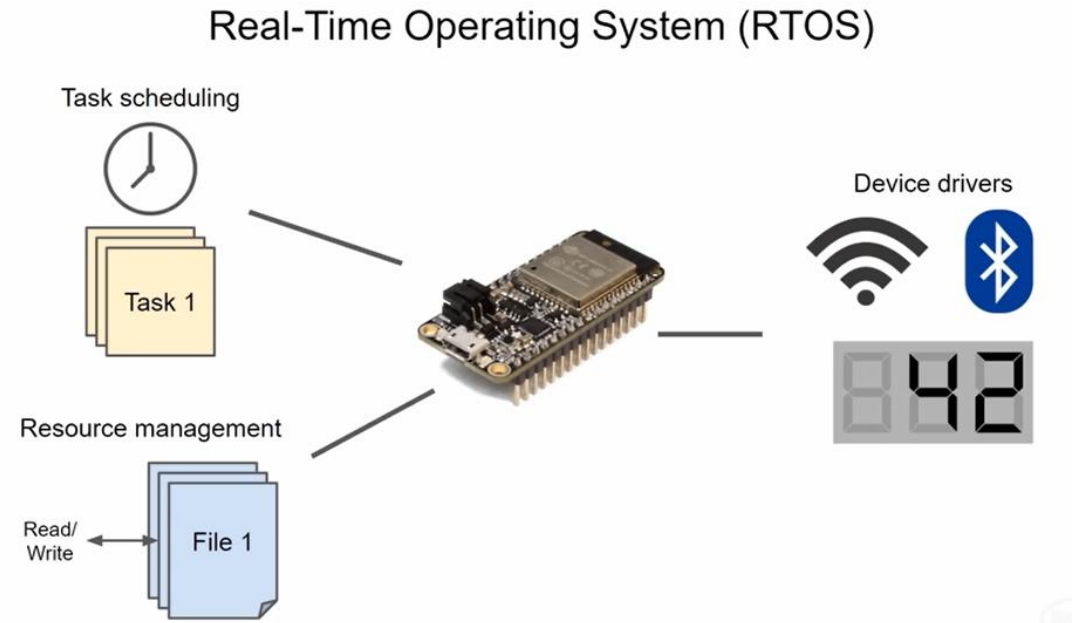

Business Layer
Business Models
System Management
HMI
Application Layer
Graphic Data Representation
Smart Applications
Middleware Layer
Decision Unit
Data Analytics
Process Information
http://www
XBee
Wi Fi
3G
4G
5G
Network Layer
Network Technologies
Data Transmission
RFID
Perception Layer
Physical Objects
Sensor & Actuators
Data Gathering

Lecture Topics

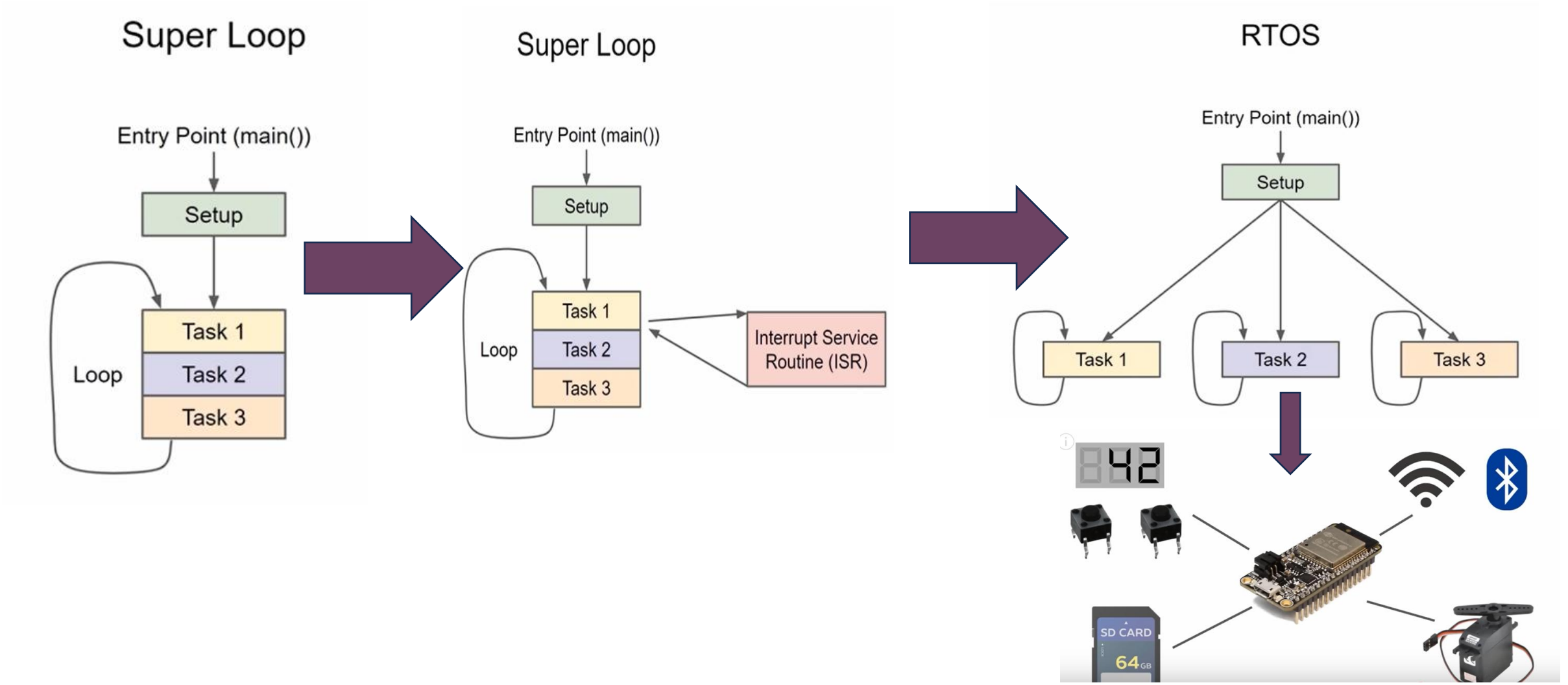
- GPOS Vs RTOS
- Why we need RTOS?
- Key Features of RTOS
- Types od RTOS

GPOS Vs RTOS


General Purpose Operating System (GPOS)
Task scheduling
Task 1
Device drivers
Resource management
Read/
Write
File 1


Real-Time Operating System (RTOS)
Task scheduling
Task 1
Device drivers
Resource management
Read/
Write
File 1

Why we need RTOS?



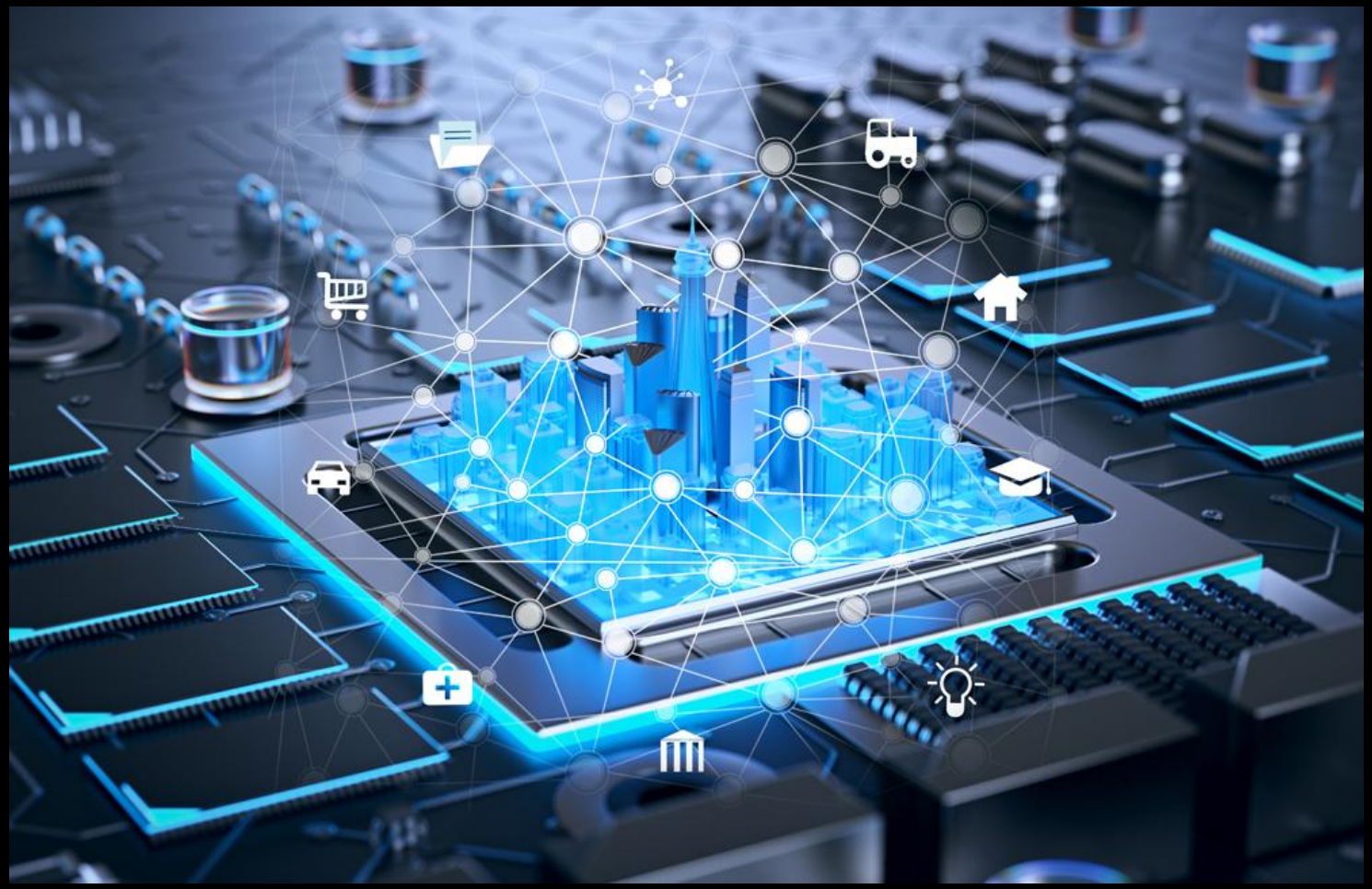
Real-Time Operating System (RTOS)

- A **Real-Time Operating System (RTOS)** is an operating system designed to handle tasks with strict timing constraints.
- It ensures that critical operations are executed within a defined time frame, making it ideal for systems where delays could lead to failure.
- RTOS is commonly used in embedded systems, industrial automation, robotics, aerospace, and medical devices.

When using RTOS

- Real-time **operating systems (RTOS)** are used in environments where a large number of events, mostly external to the computer system, must be accepted and processed in a short time or within certain deadlines. such applications are industrial control, telephone switching equipment, flight control, and real-time simulations.
- With an RTOS, the processing time is measured in tenths of seconds. This system is time-bound and has a fixed deadline. The processing in this type of system must occur within the specified constraints. Otherwise, This will lead to system failure.
- Examples of real-time operating systems are airline traffic control systems, Command Control Systems, airline reservation systems, Heart pacemakers, Network Multimedia Systems, robots, etc.

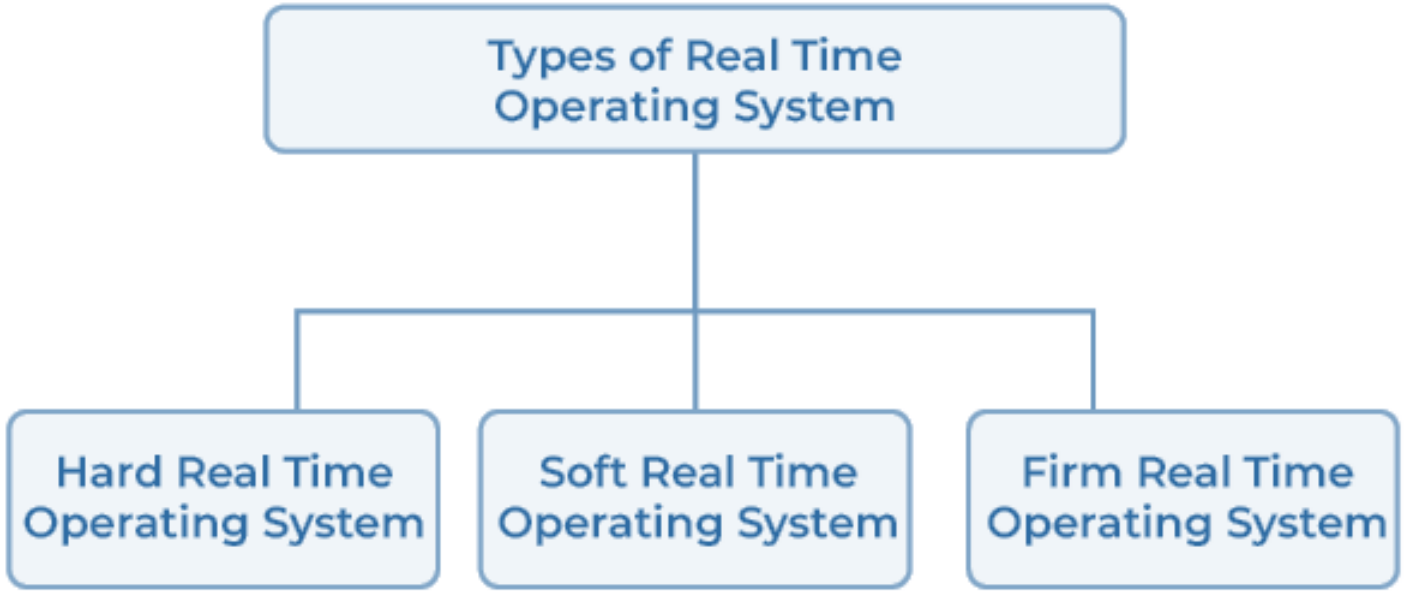
RTOS Environment



Key Features of RTOS:

- **Deterministic Behavior:** Guarantees task execution within a predictable time.
- **Task Scheduling:** Uses priority-based scheduling to ensure critical tasks run on time.
- **Multitasking Support:** Manages multiple tasks efficiently.
- **Minimal Latency:** Reduces response time to meet real-time constraints.
- **Resource Management:** Optimizes CPU, memory, and other system resources.

Types od RTOS



Hard Real-Time Operating System

- These operating systems guarantee that critical tasks are completed within a range of time.
- A system where missing a deadline leads to a **catastrophic failure**. These systems **must** complete tasks within strict time limits.
- **Characteristics:**
- Deterministic behavior (guaranteed execution time).
- Used in **critical applications** like medical devices, industrial automation, and avionics.
- **Examples:**
- **VxWorks** (Used in spacecraft & aviation).
- **RTEMS** (NASA projects, industrial control).
- **QNX** (Automotive & medical devices).
- **Use Case:**
  A **pacemaker** must deliver electrical impulses **precisely on time**. A delay could be fatal.

Soft Real-Time Operating System

- This operating system provides some relaxation in the time limit.
- A system where missing a deadline **reduces performance but doesn't cause system failure**.
- **Characteristics:**
- Best effort scheduling.
- Tasks should meet deadlines **most of the time**.
- Common in **multimedia streaming, gaming, and telecom** systems.
- **Examples:**
- **Linux with RT patches** (Real-time Linux).
- **Windows CE** (Used in embedded devices).
- **FreeRTOS** (For IoT & embedded applications).

Use Case:
A **video streaming service** should minimize buffering, but a small delay won't crash the system.

Comparison: RTOS vs. General-Purpose OS (GPOS)

Feature	**RTOS (Real-Time OS)**	**GPOS (General-Purpose OS)**
Purpose	Designed for real-time applications with strict timing constraints.	Designed for general computing tasks like desktop, mobile, and web applications.
Task Scheduling	Priority-based (Preemptive or Cooperative) for predictable execution.	Fair scheduling (e.g., Round Robin, Multilevel Queue) for balanced performance.
Determinism	Highly deterministic (guaranteed response time).	Non-deterministic (response time may vary).
Latency	Low and predictable latency.	Higher and variable latency.
Multitasking	Efficient multitasking with real-time constraints.	Supports multitasking but without strict real-time guarantees.
Resource Usage	Optimized for minimal resource consumption.	Requires more memory and processing power.
Use Cases	Embedded systems, robotics, medical devices, industrial automation.	PCs, smartphones, gaming consoles, cloud servers.
Examples	FreeRTOS, VxWorks, QNX, RTEMS.	Windows, Linux, macOS, Android.

Firm Real-time Operating System

- A system where missing a deadline **reduces system quality, but occasional misses are tolerable**.
- 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.
- **Characteristics:**
- Deadlines should be met **most of the time**.
- Some delay is acceptable, but too many missed deadlines affect performance.
- Used in **financial trading systems, robotics, and cloud computing**.
- **Examples:**
- **RTLinux** (Low-latency Linux for robotics).
- **Zephyr OS** (IoT applications with cloud integration).
- **Use Case:**
  A **stock trading system** needs fast response times, but occasional delays won't completely crash operations.

RTOS Type comparison

RTOS Type	Deadline Miss Impact	Examples	Use Cases
Hard RTOS	Catastrophic Failure	VxWorks, RTEMS, QNX	Pacemakers, Airbag Systems
Soft RTOS	Reduced Performance	FreeRTOS, RTLinux	Video Streaming, IoT Devices
Firm RTOS	Quality Reduction	Zephyr OS, RTLinux	Stock Trading, Cloud Robotics

Industrial Automation (Hard RTOS)

- **Use Case:** A factory **robotic arm** must execute precise movements to assemble products within **milliseconds** to avoid defects.
  RTOS Requirements:
- Hard real-time performance (no missed deadlines).
- High reliability & determinism.
- Support for industrial protocols (Modbus, CAN, Profibus).
- **Best RTOS Choices:**
- **VxWorks** (Used in aerospace & industrial control).
- **RTEMS** (Real-time embedded mission-critical applications).
- **QNX** (Stable and secure for industrial automation).

IoT Smart Home Devices (Soft RTOS)

- **Use Case:** A **smart thermostat** collects temperature data and adjusts heating/cooling. Occasional response delay (milliseconds) is **acceptable**.
  RTOS Requirements:
- Low power consumption.
- Internet connectivity (MQTT, Wi-Fi, Bluetooth).
- Efficient multitasking for sensors and communication.
- **Best RTOS Choices:**
- **FreeRTOS** (Lightweight, widely used in IoT).
- **Zephyr OS** (Cloud-friendly, secure).
- **Amazon FreeRTOS** (AWS integration for IoT).

More Examples

- **Medical Devices (Hard RTOS)**
- **Use Case:** A **pacemaker** must deliver electrical impulses **precisely on time** to regulate heartbeats. A failure could be fatal.
- **RTOS Requirements:**
- Predictable real-time scheduling.
- Ultra-low latency & power efficiency.
- Compliance with medical standards (FDA, IEC 62304).
- **Best RTOS Choices:**
- **VxWorks** (Certified for medical devices).
- **QNX** (Used in life-supporting systems).
- **RTEMS** (Reliable & open-source).

- **Wearable Devices (Soft/Firm RTOS)**
- **Use Case:** A **fitness tracker** collects heart rate & steps, sending data to a smartphone. Occasional delays are acceptable.
  RTOS Requirements:
- Low power consumption (battery optimization).
- Bluetooth connectivity.
- Efficient multitasking for sensor fusion.
- **Best RTOS Choices:**
- **FreeRTOS** (Optimized for low-power MCUs).
- **Zephyr OS** (Secure & scalable).
- **TinyOS** (Minimal footprint for battery devices).

Summary: RTOS Selection Guide

Use Case	Best RTOS	Real-Time Type
Industrial Automation	VxWorks, QNX, RTEMS	Hard RTOS
IoT Smart Home	FreeRTOS, Zephyr OS	Soft RTOS
Automotive	QNX, VxWorks, AUTOSAR OS	Hard RTOS
Medical Devices	VxWorks, QNX, RTEMS	Hard RTOS
Aerospace & Defense	RTEMS, VxWorks, Integrity OS	Hard RTOS
Wearables	FreeRTOS, Zephyr OS, TinyOS	Soft/Firm RTOS

Advantages

- The advantages of real-time operating systems are as follows:
- **Maximum Consumption:** Maximum utilization of devices and systems. Thus more output from all the resources.
- **Task Shifting:** Time assigned for shifting tasks in these systems is very less. For example, in older systems, it takes about 10 microseconds. Shifting one task to another and in the latest systems, it takes 3 microseconds.
- **Focus On Application:** Focus on running applications and less importance to applications that are in the queue.
- **Real-Time Operating System In Embedded System:** Since the size of programs is small, RTOS can also be embedded systems like in transport and others.
- **Error Free:** These types of systems are error-free.
- **Memory Allocation:** Memory allocation is best managed in these types of systems.

Disadvantages

- The disadvantages of real-time operating systems are as follows:
- **Limited Tasks:** Very few tasks run simultaneously, and their concentration is very less on few applications to avoid errors.
- Use Heavy System Resources: Sometimes the system resources are not so good and they are expensive as well.
- **Complex Algorithms :** The algorithms are very complex and difficult for the designer to write on.
- Device Driver And Interrupt Signals: It needs specific device drivers and interrupts signals to respond earliest to interrupts.
- **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.
- **Minimum Switching:** RTOS performs minimal task switching.

NEXT TIME

FreeRTOS