

**Experiment-2**

Operating Systems

**Program:**

**MCA**

**Lab Experiment: Comparative Study of Real-Time Operating System (RTOS), Distributed Operating System (DOS), and Mobile Operating System (MOS)**

**Objective:**

To conduct a comprehensive study of Real-Time Operating Systems (RTOS), Distributed Operating Systems (DOS), and Mobile Operating Systems (MOS), focusing on identifying types, pros and cons, challenges faced by developers, and comparing RTOS with General Purpose Operating Systems (GPOS).

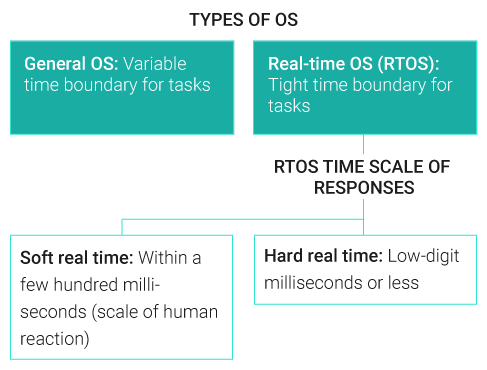
**Learning Outcomes:**

This experiment will provide a comprehensive understanding of RTOS, DOS, and MOS, their pros and cons, challenges faced by developers, and how RTOS differs from GPOS.

**To be referred by students:**

A real-time operating system (RTOS) is an operating system with two key features: predictability and determinism. In an RTOS, repeated tasks are performed within a tight time boundary, while in a general-purpose operating system, this is not necessarily so. Predictability and determinism, in this case, go hand in hand: We know how long a task will take, and that it will always produce the same result.

RTOSes are subdivided into “soft” real-time and “hard” real- time systems. Soft real-time systems operate within a few hundred milliseconds, at the scale of a human reaction. Hard real-time systems, however, provide responses that are predictable within tens of milliseconds or less.



**Characteristics of a real-time operating system**

Real-time operating systems generally have the following characteristics:

* **Small footprint.** Compared to general OSes, real-time operating systems are lightweight.
* **High performance.** RTOSes are typically fast and responsive.
* **Determinism.** Repeating inputs end in the same output.
* **Safety and security.** Safety-critical and security standards are typically the highest priority, as RTOSes are frequently used in critical systems.
* **Priority-based scheduling.** Tasks that are assigned a high priority are executed first followed by lower-priority jobs.
* **Timing information.** RTOSes are responsible for timing and providing [application programming interface](https://www.techtarget.com/searchapparchitecture/definition/application-program-interface-API)

**Part 1: Identify Various Types of RTOS**

1. **Research**:
   * Use online resources and textbooks to identify different types of RTOS.
   * Focus on categories such as Hard RTOS, Soft RTOS, and Firm RTOS.
2. **Documentation**:
   * Create a table with columns: Type of RTOS, Examples, Primary Use Cases.
   * Fill in the table with the information gathered. Example:

| **Type of RTOS** | **Examples** | **Primary Use Cases** |
| --- | --- | --- |
| Hard RTOS | VxWorks | Aerospace, Industrial Automation |
| Soft RTOS | FreeRTOS | Consumer Electronics, IoT Devices |
| Firm RTOS | QNX | Automotive, Medical Devices |

**Part 2: Identify the Pros and Cons of RTOS, DOS, and MOS**

1. **Research**:
   * Investigate the advantages and disadvantages of using RTOS, DOS, and MOS.
   * Use academic papers, technical articles, and authoritative websites for information.
2. **Documentation**:
   * Create a comparative table with columns: OS Type, Pros, Cons.
   * List the pros and cons for RTOS, DOS, and MOS. Example:

| **OS Type** | **Pros** | **Cons** |
| --- | --- | --- |
| RTOS | Deterministic response times, High reliability | Limited resources, Complex debugging |
| DOS | Scalability, Resource sharing | Network dependency, Synchronization issues |
| MOS | Mobility support, User-friendly interfaces | Fragmentation, Security risks |

**Part 3: Identify the Challenges Faced by Developers**

1. **Research**:
   * Study the common challenges faced by developers when working with RTOS, DOS, and MOS.
   * Look for case studies, developer blogs, and industry reports.
2. **Documentation**:
   * Create a detailed report highlighting the key challenges and possible solutions.
   * Structure the report with sections: RTOS Challenges, DOS Challenges, MOS Challenges. Example:

**RTOS Challenges**:

* Real-time scheduling
* Handling concurrent processes
* Ensuring minimal latency

**DOS Challenges**:

* Data consistency
* Synchronization
* Network latency

**MOS Challenges**:

* Fragmentation
* Security
* Power management

**Part 4: Difference Between GPOS and RTOS**

1. **Research**:
   * Understand the fundamental differences between General Purpose Operating Systems (GPOS) and Real-Time Operating Systems (RTOS).
   * Focus on aspects like scheduling, response time, resource management, and use cases.
2. **Documentation**:
   * Create a comparative table with columns: Feature, GPOS, RTOS.
   * Detail the differences between GPOS and RTOS for each feature. Example:

| **Feature** | **GPOS** | **RTOS** |
| --- | --- | --- |
| Scheduling | Fairness, Multi-tasking | Priority-based, Deterministic |
| Response Time | Variable, Not guaranteed | Predictable, Guaranteed |
| Resource Management | Optimized for throughput | Optimized for minimal latency |
| Use Cases | Desktop, Servers | Embedded Systems, Critical Tasks |

**Conclusion:**

* Summarize the findings from the research and documentation.
* Discuss how the different operating systems address specific needs and challenges in their respective domains.
* Highlight the key differences between GPOS and RTOS, emphasizing the importance of choosing the right OS for specific applications.

**Deliverables:**

* Completed tables and detailed report as per the documentation sections.
* Summary of findings and discussion in the conclusion section.

PART B

* (PART B: TO BE COMPLETED BY STUDENTS)
* ***(Students must submit the soft copy as per following segments as per the submission instructions.)***

| Roll No.: | Name: |
| --- | --- |
| Class: | Batch: |
| Experiment Number- | |
| Date of Experiment: | Date of Submission: |
| Grade: |  |

* **B.1 Program/Documentation with Output to be written by student**