Frequently Asked RTOS Interview Questions and Answers

Linkedin

Owner	UttamBasu
Author	Uttam Basu
Linkedin	www.linkedin.com/in/uttam-basu/

Type - Short

1. **What is an RTOS?**
   An RTOS (Real-Time Operating System) is designed to process data and respond within a strict time constraint, making it predictable and reliable for embedded applications.
2. **Explain hard and soft real-time systems.**
   Hard real-time systems must meet deadlines strictly. Soft real-time systems allow occasional deadline misses.
3. **What is task scheduling?**
   It is the mechanism by which an RTOS decides which task to run next, based on priorities or other criteria.
4. **What is preemptive scheduling?**
   In preemptive scheduling, a higher-priority task can interrupt a lower-priority task to ensure timely execution.
5. **What is cooperative scheduling?**
   Tasks voluntarily yield control, so the system depends on tasks behaving correctly.
6. **What is a context switch?**
   The process of saving the state of one task and loading the state of another.
7. **What is a semaphore?**
   A signaling mechanism used for synchronization between tasks.
8. **Types of semaphores?**
   Binary semaphore and counting semaphore.
9. **What is a mutex?**
   A mutual exclusion object that allows only one task to access a resource at a time.
10. **Difference between semaphore and mutex?**
    Mutex has ownership and is used for mutual exclusion; semaphore is for signaling and synchronization.
11. **What is priority inversion?**
    A lower-priority task holding a resource blocks a higher-priority task.
12. **How to prevent priority inversion?**
    Using priority inheritance protocols.
13. **What is deadlock?**
    A situation where two or more tasks wait indefinitely for each other's resources.
14. **What is starvation?**
    A task waits indefinitely due to being continuously preempted by higher-priority tasks.
15. **What is a task or thread?**
    A basic unit of execution in an RTOS.
16. **What are task states in RTOS?**
    Ready, Running, Blocked, Suspended.
17. **What is inter-task communication (ITC)?**
    Mechanisms like message queues, semaphores, mailboxes used for data exchange.
18. **What is a mailbox?**
    A method to send messages between tasks.

19. **What is a message queue?**
    A FIFO structure for sending messages between tasks.
20. **What is latency?**
    The delay between an event and the system's response to that event.
21. **How to reduce latency in RTOS?**
    Optimize ISR, use fast scheduling, and reduce blocking operations.
22. **What is a watchdog timer?**
    A timer that resets the system if software becomes unresponsive.
23. **What is jitter?**
    Variability in response or execution timing.
24. **What is a tick timer?**
    A timer that generates interrupts at regular intervals for task scheduling.
25. **What is an ISR (Interrupt Service Routine)?**
    A function that executes in response to a hardware interrupt.
26. **Can an ISR block or sleep?**
    No. It must be quick and should not block or delay.
27. **What is stack overflow in RTOS?**
    When a task uses more stack than allocated, leading to unpredictable behavior.
28. **What is a critical section?**
    A code block that must not be interrupted, often protected by semaphores or mutexes.
29. **What is reentrant code?**
    A function that can be safely called by multiple tasks or ISRs simultaneously.
30. **What is the RTOS kernel?**
    The core of the RTOS managing task scheduling, timing, and synchronization.
31. **What is kernel preemption?**
    The kernel allows high-priority tasks to interrupt lower-priority tasks.
32. **What is round-robin scheduling?**
    Tasks are scheduled in a rotating fashion for a fixed time slice.
33. **What is time slicing?**
    Dividing CPU time equally among tasks of the same priority.
34. **What is a real-time clock (RTC)?**
    A hardware timer that tracks actual calendar time.
35. **What is IPC (Inter-Process Communication)?**
    Mechanisms for tasks to share data and messages, e.g., queues, pipes, mailboxes.
36. **What is the difference between static and dynamic memory allocation in RTOS?**
    Static memory is allocated at compile time; dynamic memory is allocated at runtime.
37. **Why avoid dynamic memory allocation in RTOS?**
    It can lead to fragmentation and unpredictable timing.
38. **What is memory fragmentation?**
    When memory is broken into small unusable blocks due to dynamic allocation.
39. **What is a scheduler?**
    Part of the RTOS that decides which task to run next.
40. **What is priority-based scheduling?**
    Tasks are scheduled based on their priority value.

41. **What is thread-safe code?**
Code that behaves correctly when accessed by multiple tasks/threads concurrently.
42. **What is the role of the bootloader?**
It initializes hardware and loads the main application or OS.
43. **What is stack memory used for in tasks?**
For storing the task's local variables and function call context.
44. **What is the purpose of idle task?**
To run when no other task is ready to execute.
45. **What is a delay function in RTOS?**
It suspends a task for a specific period.
46. **What is tickless mode in RTOS?**
An energy-saving mode where periodic timer interrupts are reduced or disabled.
47. **What is a signal in RTOS?**
A method for one task to notify another task or ISR.
48. **Difference between embedded OS and RTOS?**
RTOS is a subtype of embedded OS with strict timing guarantees.
49. **What are hooks in RTOS?**
User-defined functions that execute at specific kernel events.
50. **Benefits of using RTOS in embedded systems?**
Deterministic timing, multitasking, modularity, and better control over hardware.
51. **How does an RTOS handle interrupt latency?**
By keeping ISR short and delegating work to tasks using deferred interrupt handling.
52. **What is deferred interrupt handling?**
Offloading interrupt processing to a lower-priority task to avoid long ISRs.
53. **What is a non-maskable interrupt (NMI)?**
A high-priority interrupt that cannot be disabled, used for critical events.
54. **Explain real-time scheduling policies.**
Common policies include Rate Monotonic Scheduling (RMS) and Earliest Deadline First (EDF).
55. **What is Rate Monotonic Scheduling?**
A fixed-priority scheduling where shorter period tasks get higher priority.
56. **What is Earliest Deadline First (EDF)?**
A dynamic scheduling method where tasks with the nearest deadline are prioritized.
57. **What is stack pointer and how is it used in context switching?**
It tracks the top of the stack and is saved/restored during task switches.
58. **What happens during task creation in an RTOS?**
A new stack is allocated, task control block initialized, and added to the ready list.
59. **What is a task control block (TCB)?**
A data structure storing information about a task (state, priority, stack, etc.).
60. **What is task deletion?**
Removing a task from the system and reclaiming its resources.
61. **What is event flag or event group?**
A group of flags used to synchronize tasks based on multiple events.

62. **What is a software timer in RTOS?**
    A timer maintained in software that invokes a callback when it expires.
63. **What is the difference between software and hardware timers?**
    Hardware timers are based on physical counters, software timers are managed in code.
64. **What is priority ceiling protocol?**
    Prevents priority inversion by temporarily raising the priority of a resource-owning task.
65. **What is memory protection in RTOS?**
    Ensures tasks can only access allocated memory, preventing corruption.
66. **What is the difference between process and thread in RTOS?**
    Threads share memory; processes are isolated.
67. **What is load balancing in RTOS?**
    Distributing workload evenly across processors in SMP systems.
68. **Explain RTOS porting.**
    Adapting the RTOS to work with a new processor or hardware architecture.
69. **What is reentrancy vs thread-safety?**
    Reentrancy ensures safe recursion; thread-safety ensures concurrent access is safe.
70. **What is dual-mode operation in RTOS?**
    Operating in privileged (kernel) and user modes for security.
71. **What is the use of yield() function in RTOS?**
    Allows a task to voluntarily relinquish the CPU.
72. **What is a kernel object?**
    Structures used for synchronization and communication (semaphores, queues, etc.).
73. **What are real-time constraints?**
    Conditions like deadline, response time, and determinism.
74. **How to test real-time constraints?**
    Using profiling, simulation, and timing analysis tools.
75. **What is task aging?**
    Increasing task priority over time to prevent starvation.
76. **What is priority inheritance?**
    Temporarily boosting a low-priority task's priority to avoid inversion.
77. **What is task blocking?**
    A task is put to sleep while waiting for an event or resource.
78. **What is RTOS footprint?**
    Total memory (code + data) required to run the RTOS.
79. **What is preemption threshold?**
    A task can prevent preemption by others below a threshold priority.
80. **What is thread join and detach?**
    Join waits for thread to finish; detach runs independently.
81. **What is memory leak and how to detect it?**
    Memory not freed after use; tools like valgrind can help detect it.
82. **Can an RTOS run on bare-metal systems?**
    Yes. Most RTOSes are designed to run directly on hardware.
83. **What is the maximum number of tasks an RTOS can support?**
    Depends on memory and RTOS design, often hundreds or more.

84. **What is a tickless idle mode?**
    A low-power mode where periodic system ticks are disabled.
85. **What is CPU utilization in RTOS?**
    Measure of CPU time used vs idle time.
86. **What is power management in RTOS?**
    Managing CPU states (sleep, deep sleep) to reduce power consumption.
87. **What is RTOS trace tool?**
    Software used to monitor and analyze RTOS behavior during runtime.
88. **What is a real-time guarantee?**
    Assurance that deadlines will be met under all defined conditions.
89. **What is a critical path in real-time tasks?**
    The longest sequence of dependent tasks determining overall response time.
90. **What is slack time?**
    Time between task completion and its deadline.
91. **How does an RTOS manage heap memory?**
    With allocators like first-fit, best-fit, or fixed-size blocks.
92. **What is a BSP (Board Support Package)?**
    Set of drivers and configurations allowing RTOS to interface with hardware.
93. **What is inter-core communication in SMP RTOS?**
    Methods like shared memory or mailboxes for task coordination across cores.
94. **What is the role of scheduler lock?**
    Temporarily disables context switching for critical sections.
95. **What is ISR nesting?**
    Allowing one interrupt to be interrupted by another of higher priority.
96. **What are system services in RTOS?**
    Core functions like task creation, timing, IPC, and synchronization.
97. **What is the difference between ISR and DSR (Deferred Service Routine)?**
    ISR handles immediate needs; DSR does deferred, longer processing.
98. **What is kernel panic?**
    Fatal error in the RTOS kernel requiring a system reset.
99. **What is cooperative multitasking's main drawback?**
    A task can monopolize CPU if it doesn't yield.
100. **What metrics are used to evaluate an RTOS?**
    Latency, jitter, context switch time, determinism, footprint, power efficiency.
101. **What is an embedded RTOS?**
    An RTOS optimized for embedded systems with limited resources and strict real-time constraints.
102. **Why is determinism important in RTOS?**
    It ensures tasks always complete within predictable timing limits.
103. **What is bounded latency?**
    The maximum delay between a stimulus and the RTOS response, which must be known and predictable.
104. **What is thread starvation?**
    When lower-priority threads never get scheduled due to constant preemption by higher ones.

105. **Explain the difference between scheduling latency and interrupt latency.**
Scheduling latency is the time to switch tasks; interrupt latency is the time from interrupt to ISR execution.
106. **What is the minimum stack size per thread?**
It depends on the function calls and local variables used; estimated during design/testing.
107. **What is a tick interrupt?**
A timer interrupt that triggers the RTOS scheduler at fixed intervals.
108. **What is a dormant task?**
A task that has been created but is not ready, running, or waiting — essentially idle or terminated.
109. **What is a lightweight RTOS?**
An RTOS designed for minimal footprint and low-power devices, such as FreeRTOS.
110. **What is a non-preemptible section?**
A code section where preemption is disabled to protect shared resources.
111. **What is a cyclic executive system?**
A non-RTOS method where tasks are called in a fixed sequence, often in time slots.
112. **What is cooperative scheduling's biggest risk?**
A task may hog CPU time, causing missed deadlines.
113. **How do you profile an RTOS application?**
Using trace tools to analyze task execution, timing, CPU usage, and events.
114. **What is an alarm in RTOS?**
A timed action or callback invoked after a specific delay.
115. **What is job scheduling in real-time systems?**
Assigning processor time to jobs (task instances) based on deadlines and priorities.
116. **What is a reentrant ISR?**
An ISR that can be interrupted and safely re-entered without data corruption.
117. **What is the heap used for in RTOS?**
Dynamic memory allocation at runtime (if used).
118. **What's the difference between an idle thread and an idle loop?**
An idle thread is managed by the RTOS; an idle loop is manually implemented.
119. **Can tasks call ISRs?**
No. ISRs are triggered by hardware and managed by the CPU interrupt controller.
120. **What is shared memory communication?**
Tasks use common memory regions to exchange data, with synchronization mechanisms.
121. **What is worst-case execution time (WCET)?**
The maximum time a task may take to execute under worst-case conditions.
122. **Why avoid delays in ISRs?**
Delays block lower-priority interrupts and increase system response times.
123. **What is tick granularity?**
The smallest unit of time the scheduler can measure or act upon.
124. **What is a round-robin with priority scheduling?**
Equal-priority tasks share time slices, but higher-priority tasks preempt them.
125. **What is an exception handler?**
A function invoked when a software exception (e.g., divide by zero) occurs.

126. **What is latency budgeting?**
Planning the time taken for each task or interrupt to meet system deadlines.
127. **What is DMA and how does it help RTOS performance?**
Direct Memory Access transfers data independently of the CPU, reducing load.
128. **What is a system tick count?**
A continuously increasing counter representing elapsed ticks since startup.
129. **What is a memory pool in RTOS?**
A preallocated memory block divided into fixed-size chunks for fast allocation.
130. **What is deterministic behavior?**
The guarantee that the system will always respond within expected time limits.
131. **What is an I/O bound task?**
A task that mostly waits for input/output operations to complete.
132. **What is the difference between asynchronous and synchronous events?**
Asynchronous events occur independently; synchronous events occur in a defined sequence.
133. **What is the difference between delay and sleep in RTOS?**
Delay pauses a task for a specific time; sleep may imply suspension until event occurrence.
134. **What is static task allocation?**
Defining task parameters and memory at compile time rather than runtime.
135. **Why use real-time trace logging?**
To debug, analyze, and optimize RTOS timing and behavior during development.
136. **What are OS services in RTOS?**
Built-in features like task creation, delay, IPC, and memory management.
137. **What is a global lock?**
A lock that affects multiple components or the entire system, often for debugging.
138. **What is a race condition?**
An error caused when two tasks access shared data concurrently without proper synchronization.
139. **What is the difference between polling and interrupt-based systems?**
Polling repeatedly checks for events; interrupts react instantly when events occur.
140. **What is a priority queue?**
A data structure used by the scheduler to sort tasks by priority.
141. **How do you choose the tick frequency?**
It balances timing accuracy with CPU overhead — common values range from 1ms to 10ms.
142. **What is execution context in RTOS?**
It includes registers, stack, and CPU state of a task or ISR.
143. **What is a spinlock and when is it used?**
A lock where a task repeatedly checks a variable until it's available — useful in multiprocessor systems.
144. **What are system invariants?**
Conditions that must always hold true for the RTOS to work correctly.
145. **What are some examples of RTOS?**
FreeRTOS, VxWorks, Micrium, Zephyr, ThreadX, QNX, etc.
146. **What is a real-time deadline miss?**
When a task fails to finish before its deadline.

147. **What is safe state recovery?**
Returning the system to a known safe state after a fault or crash.
148. **What is a critical fault handler?**
A routine invoked on unrecoverable errors, often performing a reset or log.
149. **What are application hooks in RTOS?**
User-defined callbacks triggered by RTOS events like task create/delete.
150. **What is a hybrid RTOS kernel?**
A kernel combining real-time and general-purpose features for flexibility.
151. **What is a safety-critical RTOS?**
An RTOS used in systems where failure can cause loss of life or severe damage (e.g., automotive, medical).
152. **What are examples of safety-certified RTOSes?**
QNX, VxWorks Cert Edition, Integrity RTOS, and SafeRTOS.
153. **What is ARINC 653?**
An RTOS specification for avionics systems defining time and space partitioning.
154. **What is time partitioning?**
Allocating CPU time in fixed slots to ensure deterministic behavior across partitions.
155. **What is space partitioning?**
Each task or process is isolated in its memory space for fault containment.
156. **What is memory protection unit (MPU)?**
A hardware feature that enforces access restrictions to memory regions.
157. **How does an RTOS support mixed-criticality systems?**
By isolating tasks using time and space partitioning and enforcing strict scheduling policies.
158. **What is a hypervisor in RTOS context?**
A layer that allows multiple OS instances or RTOSes to run on the same hardware.
159. **What is a guest OS?**
An OS that runs under the control of a hypervisor.
160. **Difference between RTOS and hypervisor?**
An RTOS manages real-time tasks; a hypervisor manages multiple OS environments.
161. **What is an A/B system upgrade in embedded RTOS?**
Running two firmware images (A and B) for seamless updates and rollback.
162. **What is boot time optimization in RTOS?**
Minimizing the startup time to meet real-time constraints or improve UX.
163. **What is OS-aware debugging?**
A debugger that understands RTOS tasks, semaphores, queues, and can show RTOS states.
164. **What is deterministic scheduling?**
Scheduling with predictable, analyzable behavior ensuring deadlines are met.
165. **What is a time-triggered system?**
A system where tasks run at predetermined times instead of reacting to events.
166. **What is jitter control in RTOS?**
Minimizing variations in task start times for consistent behavior.
167. **What is multicore RTOS scheduling?**
RTOS assigns and balances tasks across multiple CPU cores.

168. **What is asymmetric multiprocessing (AMP)?**
Each core runs its own OS or RTOS instance independently.
169. **What is symmetric multiprocessing (SMP)?**
Multiple cores share the same OS and memory, coordinated by the RTOS.
170. **How does RTOS support SMP?**
Via core affinity, load balancing, and locking mechanisms.
171. **What is interrupt storming?**
Excessive interrupt frequency causing CPU to be overloaded.
172. **How to mitigate interrupt storming?**
Using interrupt throttling, masking, and deferred handling.
173. **What is a trace buffer?**
A memory area used to store execution logs and events for analysis.
174. **What is kernel instrumentation?**
Modifying the RTOS kernel to provide runtime diagnostics and logging.
175. **What is task instrumentation?**
Adding hooks to tasks to track their behavior and performance.
176. **What is a heartbeat task?**
A task that runs periodically to signal system health.
177. **What is time drift in real-time systems?**
Gradual deviation of system clock from real-world time.
178. **How is time synchronization achieved in distributed RTOS systems?**
Using protocols like NTP, IEEE 1588 PTP, or GPS-based sync.
179. **What is CAN-based task triggering?**
Using CAN messages to activate tasks in automotive systems.
180. **What is rate monotonic analysis (RMA)?**
A method for verifying schedulability in fixed-priority RTOS systems.
181. **What is schedulability test?**
Analyzing if all tasks can complete within deadlines given system constraints.
182. **What is slack stealing?**
Using unused time (slack) from low-priority tasks to help other tasks.
183. **What is sporadic task?**
A task that occurs irregularly but within a known minimum inter-arrival time.
184. **What is an aperiodic task?**
A task with no regular pattern and unpredictable arrival times.
185. **What are the common IPC mechanisms in RTOS?**
Semaphores, message queues, mailboxes, shared memory, events.
186. **What is stack watermarking?**
Technique to determine peak stack usage by filling stack with known values.
187. **What is lock-free programming in RTOS?**
Concurrency control without mutexes or semaphores, using atomic operations.
188. **What is priority boosting?**
Temporarily increasing a task's priority to prevent starvation.
189. **What is power-aware scheduling?**
Scheduling that considers task energy requirements and optimizes CPU usage.

190. **What is tick suppression in RTOS?**
Disabling unnecessary system ticks when idle to save power.
191. **What are system hooks?**
User-defined functions called during events like context switches or task deletions.
192. **What is CPU hogging?**
When a task uses too much CPU time, potentially blocking others.
193. **What is a trusted execution environment (TEE)?**
A secure RTOS partition that runs sensitive code isolated from the main OS.
194. **What is ISO 26262 in RTOS context?**
A standard for functional safety in automotive systems, often requiring RTOS certification.
195. **What is DO-178C for RTOS?**
A safety-critical software standard for aviation, often requiring certifiable RTOSes.
196. **What is task determinism vs system determinism?**
Task determinism refers to individual task timing; system determinism is the whole system's predictability.
197. **What is fail-safe vs fail-operational RTOS design?**
Fail-safe: safe shutdown on error; fail-operational: continue functioning safely even after failures.
198. **What is context-aware scheduling?**
Scheduler adapts behavior based on task history, usage, or external signals.
199. **What is temporal isolation?**
Guaranteeing one task's timing behavior is unaffected by others.
200. **What are the advantages of a modular RTOS kernel?**
Customizability, smaller footprint, and easier certification.
201. **What is power-aware RTOS design?**
Designing the RTOS to actively manage and reduce power consumption based on system activity.
202. **How does an RTOS support energy harvesting systems?**
By allowing ultra-low power modes, long sleep cycles, and asynchronous wake-up support.
203. **What is an event-driven RTOS?**
An RTOS where execution is primarily triggered by external/internal events, not just time.
204. **What is the difference between soft timers and hard timers?**
Soft timers use RTOS tick count; hard timers are based on hardware peripherals.
205. **What is asynchronous wakeup in RTOS?**
Waking up the system from sleep mode based on external interrupts or events.
206. **What is RTOS instrumentation?**
Adding probes or hooks for capturing internal kernel events for debugging and profiling.
207. **What are trace probes?**
Mechanisms to collect execution and timing data during runtime for offline analysis.
208. **What is the benefit of using static analysis in RTOS-based systems?**
Helps detect errors like race conditions, deadlocks, and stack overflows at compile-time.
209. **What is code coverage testing?**
Verifies which parts of the code were executed during testing to ensure complete validation.

210. **What is a system health monitor in RTOS?**
A task or component that tracks CPU load, memory usage, and responsiveness of other tasks.
211. **What is a watchdog feeding interval?**
The time window within which the system must reset the watchdog to avoid system reset.
212. **What is tick alignment?**
Synchronizing the system tick to specific time boundaries to reduce jitter.
213. **What is dynamic priority adjustment?**
Modifying task priorities at runtime based on system behavior or resource usage.
214. **What is cooperative round-robin scheduling?**
Equal-priority tasks share CPU time but must yield control voluntarily.
215. **What is frequency scaling in RTOS?**
Dynamically adjusting CPU frequency to reduce power consumption.
216. **What is an adaptive scheduler?**
A scheduler that changes behavior based on runtime conditions like task load or energy.
217. **What is an overrun in real-time systems?**
When a task exceeds its allocated execution time, potentially disrupting the schedule.
218. **What is an underrun?**
When a task finishes too early, possibly leading to wasted CPU cycles.
219. **What is non-blocking communication?**
Task continues execution without waiting for the communication to complete.
220. **What is the purpose of DMA in RTOS systems?**
Offloads memory/data transfer work from the CPU, reducing latency and freeing cycles.
221. **What is a reentrant mutex?**
A mutex that allows the same task to lock it multiple times safely.
222. **What is error propagation in RTOS?**
When a task error leads to cascading failures in other parts of the system.
223. **What is isolation kernel architecture?**
A kernel design that separates critical and non-critical functions for reliability.
224. **What is static scheduling in RTOS?**
Task execution order is predefined and does not change during runtime.
225. **What is real-time garbage collection?**
A memory management technique used without breaking real-time guarantees.
226. **What is slack reclamation?**
Utilizing unused time between task completions to execute other tasks or maintenance routines.
227. **What is a sporadic server in RTOS?**
A scheduling strategy to manage aperiodic tasks within real-time constraints.
228. **What is cache coherency in SMP RTOS?**
Ensuring that all cores see the same view of memory when using CPU caches.
229. **What is temporal decoupling?**
Design where producer and consumer tasks don't need to run at the same rate or time.
230. **What is a wake lock?**
A mechanism to keep the CPU awake during critical operations.
231. **What is dynamic load balancing in RTOS?**
Moving tasks between cores at runtime to evenly distribute workload.

232. **What is fault containment in RTOS?**
Isolating faults so that they do not affect the rest of the system.
233. **What is a kernel panic handler?**
Handles unrecoverable errors by logging and resetting or halting the system.
234. **What is temporal redundancy in RTOS?**
Re-executing tasks periodically to increase reliability in fault-tolerant systems.
235. **What is spatial redundancy in RTOS?**
Running duplicate tasks on different cores or systems for fault detection.
236. **What is a fail-safe timer?**
A backup timer to ensure operations complete in a bounded time even during failures.
237. **What is software fault tolerance in RTOS?**
Techniques like watchdogs, retries, and exception handlers to recover from software bugs.
238. **What is a context overflow?**
When the data saved during a context switch exceeds allocated storage space.
239. **What is queue starvation?**
Lower-priority tasks never getting access to a shared queue due to constant higher-priority usage.
240. **What is real-time virtualization?**
Running virtual machines with real-time characteristics using RTOS/hypervisor techniques.
241. **What is bootloader-RTOS handoff?**
Transition process where the bootloader initializes and passes control to the RTOS.
242. **What is system tick drift and how is it corrected?**
Timekeeping error in the tick counter, corrected via periodic synchronization.
243. **What is a supervisor call (SVC)?**
A software interrupt used to request kernel-level services from user-mode tasks.
244. **What is the difference between spinlocks and mutexes?**
Spinlocks are busy-waiting; mutexes block the task and release the CPU.
245. **What is task instrumentation overhead?**
The CPU and memory cost of collecting runtime diagnostics.
246. **What is an assertion in RTOS?**
A runtime check that halts execution if a condition is violated, useful during debugging.
247. **What is a finite state machine (FSM) in RTOS applications?**
A model where task behavior is represented by discrete states and transitions.
248. **What is hot-patching in embedded RTOS?**
Updating code without rebooting or halting the system.
249. **What is zero-copy communication?**
Sharing data buffers directly between producers and consumers to avoid copying.
250. **What is a condition variable in RTOS?**
A synchronization primitive that blocks a task until a condition is true, often used with mutexes.