



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Sistemas Operativos

Profesor(a): Ing. Gunnar Eyal Wolf Iszaevich
Semestre 2025-1

Exposición

Nombre Trabajo
Sistemas Operativos de
Tiempo Real

Grupo: 6

Alumno:

Nava Benítez David Emilio
Tavera Castillo David Emmanuel

Introducción

Un Sistema Operativo en Tiempo Real (RTOS) es un sistema operativo especializado que procesa datos y realiza operaciones dentro de límites de tiempo específicamente definidos.

Mientras un sistema operativo tradicional prioriza la eficiencia y el manejo de múltiples tareas, un RTOS prioriza el determinismo y los tiempos de respuesta garantizados.

Un sistema operativo de uso general no está diseñado para satisfacer las exigencias de los sistemas integrados o de seguridad crítica, que requieren tiempos de respuesta constantes a entradas de múltiples fuentes, como cámaras, radares y lidars.

Los RTOS se basan en eventos y son preventivos. En otras palabras, el sistema operativo puede supervisar la prioridad relativa de las tareas en competencia y modificar sus prioridades. La forma más rápida de definir un RTOS es que estos son Sistemas Operativos que se utilizan para aplicaciones específicas con algunas restricciones de respuestas en el tiempo.

La mayoría de los RTOS ofrecen similitudes en funciones de uno de propósito general, pero para un RTOS es importante cumplir su propósito en un plazo definido y sin retraso, ya que dependiendo del contexto hasta un microsegundo es vital. Pongámonos en retrospectiva para entender esto: Los RTOS son utilizados en la industria médica donde el tiempo es muy importante; por lo tanto es indispensable no perder tiempo, ¿qué pasaría si por cuestiones del Sistema Operativo que maneja un marcapasos introduce una latencia inesperada? Conlleva a que el impulso eléctrico llegase tarde, poniendo en riesgo la vida del paciente. Para ello están los RTOS ya que asegura que las tareas de alta prioridad, como la generación del pulso eléctrico, se ejecuten de inmediato y sin interrupciones inesperadas en un tiempo garantizado.

Componentes de un RTOS

- Gestión de tareas

Las tareas se gestionan mediante un planificador, que decide qué tarea se ejecuta y cuándo, dependiendo de su prioridad y estado.

Los RTOS pueden cambiar de tarea rápidamente, a menudo en tres microsegundos o menos. Esta transferencia de tareas garantiza que los procesos críticos se completen a tiempo, lo que mejora la eficiencia general del sistema. Centrarse en la ejecución de las tareas en cola más importantes y las aplicaciones críticas puede contribuir aún más a su eficacia.

- Temporizadores

Los temporizadores en un RTOS son fundamentales para la gestión de tareas, la sincronización y la planificación. Se utilizan para realizar operaciones críticas con precisión temporal.

- **Semáforos y Mutex**

En un RTOS los semáforos y mutex están diseñados para seguir un comportamiento determinista, priorizando el cumplimiento de plazos estrictos en tareas críticas mediante mecanismos como pueden ser la herencia de prioridad e inversión de prioridades. Lo que nos asegura que los recursos se gestionen eficientemente sin afectar el tiempo real de ejecución.

En contraste, en un sistema operativo tradicional, estos mecanismos son más genéricos, enfocados en la eficiencia y la coordinación multitarea sin preocuparse por tiempos críticos, lo que los hace menos adecuados para aplicaciones donde el tiempo de respuesta es vital.

- **Gestión de memoria**

En RTOS, la memoria se divide en diferentes secciones, como la pila, heap y los segmentos de datos. El segmento de pila se utiliza para almacenar variables locales e información de llamadas a funciones. Crece hacia abajo en la memoria y requiere una gestión cuidadosa para evitar errores de desbordamiento o subdesbordamiento. Por otro lado, el segmento heap asigna memoria dinámicamente para variables u objetos durante la ejecución del programa.

Para gestionar estos segmentos eficientemente, RTOS emplea diversas técnicas, como algoritmos de asignación dinámica (como el primer ajuste o el mejor ajuste) para asignar memoria del segmento del heap.

Es importante gestionar de manera adecuada el uso de la memoria en un RTOS, debido a que estos sistemas operativos es fundamental que sean deterministas (lo que supone tener certeza de lo que sucederá en el tiempo con dicho sistema). En este aspecto, un RTOS maneja la memoria de dos maneras principalmente, de manera estática y dinámica (tal cual un Sistema Operativo sin restricciones de tiempo)

- **Atención de interrupciones**

La atención de interrupciones en un sistema operativo en tiempo real (RTOS) se realiza mediante una solicitud de interrupción (IRQ) y una rutina de servicio de interrupción (ISR).

- **Solicitud de interrupción (IRQ)**

Es una señal que indica al RTOS que debe detener la tarea actual y determinar la siguiente acción.

Ayuda a priorizar tareas y a gestionar múltiples interrupciones.

- **Rutina de servicio de interrupción (ISR)**

Implementa las funciones de sincronización de un RTOS.

Maneja las interrupciones generadas por el reloj de hardware.

Hablando específicamente de las interrupciones de hardware, debemos saber que los microcontroladores (que usan mayoritariamente RTOS) tienen periféricos integrados que

pueden generar *interrupciones de hardware*, estos pueden ser simples temporizadores que acaban su tiempo o un botón que es presionado por un usuario. Importante entender que para un RTOS una interrupción de hardware tendrá la más alta prioridad que cualquier otra tarea o proceso en ejecución aun cuando dicha tarea o proceso tenga su propia prioridad y por lo tanto dicha interrupción de hardware obligará al procesador a ejecutar la rutina de servicio de interrupción que está asociada a dicho componente.

Una vez atendida la interrupción de hardware el sistema retomará las tareas que ignora por atender dicha interrupción pero (y un grande pero) puede que regrese a la tarea que estaba ejecutando antes de la irrupción o a alguna otra dependiendo su prioridad. Por ello, en un RTOS se tiene la posibilidad de que para ciertas tareas se deshabilite una o todas las interrupciones, pero esto genera un problema ya que debemos estar conscientes de que tipo de tarea va a ignorar las interrupciones y esa deshabilitación debe hacerse con moderación y aún más si se necesita hacer para proteger algún recurso compartido crítico.

Conclusiones

Los RTOS son componentes esenciales para aplicaciones donde el tiempo y la precisión son factores determinantes. Su diseño eficiente, junto con las nuevas tendencias tecnológicas, aseguran que sigan siendo una herramienta clave en el desarrollo de sistemas embebidos y otras aplicaciones críticas en el futuro. El auge tecnológico para satisfacer las necesidades que requieren de eficiencia y sin retrasos hacen que los RTOS sean un tipo de sistema operativo que va a perdurar por el largo curso de la computación y sin duda alguna seguirá evolucionando en el transcurso de los años.

El hecho que los RTOS puedan parecer algo completamente diferente a un SO convencional no los hace para nada diferentes, ya que los sistemas en tiempo real toman la base de lo que son desde los sistemas de propósito general. Sin embargo, lo que distingue a un RTOS es su capacidad para cumplir estrictamente con los plazos de tiempo establecidos, un requisito indispensable en su diseño. Es necesario entender a un sistema operativo en tiempo real como una herramienta que gestiona el "aquí y ahora". ¿Qué significa esto? Que un RTOS está diseñado para operar bajo el principio de no desperdiciar tiempo en actividades críticas y garantizar que los procesos importantes se ejecuten en el tiempo establecido. Este principio constituye su razón de ser y justifica su existencia (en la mayoría de ocasiones).

Referencias

Aptiv. (2024, 6 diciembre). *¿Qué es un sistema operativo en tiempo real?*

Aptiv. <https://www.aptiv.com/es/tendencias/art%C3%ADculo/que-es-un-sistema-operativo-en-tiempo-real>

Zeifman, I. (2024, 9 junio). *What Is RTOS, How It Works, and 9 RTOS Platforms to Know*. Sternum IoT.

<https://sternumiot.com/iot-blog/crash-course-introduction-to-real-time-operating-system-rtos/>

Embien Technologies. *Using Interrupts for context switching in RTOS*. (s. f.). Embien.

<https://www.embien.com/blog/using-interrupts-for-context-switching-in-rtos>

Real-Time Operating System (RTOS): Working and Examples | Spiceworks. (2025, 10 marzo). Spiceworks Inc.

<https://www.spiceworks.com/tech/hardware/articles/what-is-rtos/>

Barney, N., & Gillis, A. S. (2024, 1 octubre). *What is a real-time operating system (RTOS)?* Search Data Center.

<https://www.techtarget.com/searchdatacenter/definition/real-time-operating-system>

FreeRTOS documentation - *FreeRTOS™*. (s. f.).

<https://www.freertos.org/Documentation/00-Overview>

DigiKey. (2021, 25 enero). *Introduction to RTOS Part 4 - Memory Management* | Digi-Key Electronics [Video]. YouTube. <https://www.youtube.com/watch?v=Qske3yZRW5I>

DigiKey. (2021a, enero 4). *Introduction to RTOS Part 1 - What is a Real-Time Operating System (RTOS)?* | Digi-Key Electronics [Video]. YouTube.

<https://www.youtube.com/watch?v=F321087yYy4>

Ing. Muro, Gustavo. (s. f.). *Introducción a FreeRTOS*. Universidad Nacional de Rosario. FACULTAD DE CIENCIAS EXACTAS, INGENIERÍA Y AGRIMENSURA DEPARTAMENTO DE SISTEMAS E INFORMÁTICA

Index of /docencia/TiempoReal/Recursos/temas. (s. f.).

<http://www.isa.uniovi.es/docencia/TiempoReal/Recursos/temas/sotr.pdf>