

# Apuntes sistemas en tiempo real

Roberto Castellor Morant

24-02-17

## 1. Introducción a los Sistemas de Tiempo Real

### 1.1. Definición sistema en tiempo real

Cualquier sistema en el que el tiempo en el que se produce la salida es significativo. Esto generalmente es porque la entrada corresponde a algún movimiento en el mundo físico, y la salida esta relacionada con dicho movimiento. El intervalo entre el tiempo de entrada y el de salida debe ser lo suficientemente pequeño para una temporalidad aceptable.

Cualquier actividad o sistema de proceso de información que tiene que responder a un estímulo de entrada generado externamente con un retardo finito y especificado.

La corrección de un sistema en tiempo real no depende sólo del resultado lógico de la computación, sino tambien del tiempo en el que se producen los resultados.

### 1.2. Tipos de sistemas en tiempo real

- Hard (estrictos) Son los sistemas en los que es absolutamente imperativo que las respuestas se produzcan dentro del tiempo límite especificado.
- Soft (no estrictos) Los tiempos de respuesta son importantes pero el sistema seguirá funcionando correctamente aunque los tiempos límite no se cumplan ocasionalmente.

### 1.3. Ejemplos de sistemas en tiempo real

### 1.4. Características de sistemas en tiempo real

Los sistemas en tiempo real tienen unas características específicas que los definen.

1. Funcionalidades en tiempo real
2. Control concurrente de sistemas separados
3. Programación de bajo nivel
4. Soporte a computación numérica
5. Grandes y complejos

6. Extremadamente fiables y seguros
7. Implementación eficiente y entorno de ejecución

## 2. Fiabilidad y tolerancia a fallos

Este tema, junto con el tema 3, trata de la producción de componentes de software fiables. Aunque se realizan consideraciones sobre la prevención de fallos, la atención se dedica principalmente a la tolerancia a fallos. Se condirean las técnicas de recuperación de errores hacia adelante y hacia atrás. El manejo de excepciones se estudia en el siguiente tema.

Causas que pueden propiciar el fallo de un sistema de tiempo real:

1. Especificación inadecuada.
2. Defectos provocados por errores de diseño.
3. Defectos provocados por fallos en componentes del procesador.
4. Defectos provocados por interferencias transitorias o permanentes en el subsistema de comunicaciones.

Los errores relacionados con el diseño o la especificación son difíciles de prever mientras que los errores provocados por fallos son en cierto modo predecibles. Los lenguajes de programación de tiempo real tienen que ser altamente fiables.

### 2.1. Fiabilidad, fallos y defectos

Fiabilidad: una medida del éxito con el que el sistema se ajusta a alguna especificación definitiva de su comportamiento.

Fallo del sistema: Cuando el comportamiento de un sistema se desvía del especificado para él, se dice que es un fallo.

Se pueden distinguir tres tipos de fallos:

1. Fallos transitorios, Un fallo transitorio comienza en un instante de tiempo concreto, se mantiene en el sistema durante algún periodo, y luego desaparece. Ejemplos de este tipo de fallos se dan en componentes hardware en los que se produce una reacción adversa a una interferencia externa, como la producida por un campo eléctrico o por radioactividad. Después de que la perturbación desaparece, lo hace también el fallo (aunque no necesariamente el error inducido). Muchos de los fallos en los sistemas de comunicación son transitorios.
2. Fallos permanentes Los fallos permanentes comienzan en un instante determinado y permanecen en el sistema hasta que son reparados; es el caso, por ejemplo, de un cable roto o de un error de diseño de software.
3. Fallos intermitentes Son fallos transitorios que ocurren de vez en cuando. Un ejemplo es un componente hardware sensible al calor, que funciona durante un rato, deja de funcionar, se enfría, y entonces comienza a funcionar de nuevo

## 2.2. Modos de fallo

Se pueden identificar dos dominios generales de modos de fallo:

1. Fallos de valor, el valor asociado con el servicio es erróneo.
2. Fallos de tiempo, el servicio se completa a destiempo.

Las combinaciones de fallos de valor y de tiempo se denominan fallos arbitrarios.

Un fallo de valor fuera del rango esperado para el servicio se denomina error de límites, son fallos fácilmente reconocibles.

Los fallos en el dominio del tiempo se pueden englobar en:

1. Demasiado pronto, el servicio es entregado antes de lo requerido.
2. Demasiado tarde, el servicio se entrega después de lo requerido, se puede hablar de error de prestaciones.
3. Infinitamente tarde, el servicio nunca es entregado, fallo de omisión.

Esta clasificación se puede ampliar con fallos de encargo o improvisación cuando el servicio es entregado sin ser esperado.

Dada la clasificación de fallos se puede definir algunas suposiciones respecto al modo en que los sistemas pueden fallar:

1. Fallo descontrolado, un sistema que produce fallos arbitrarios tanto en el dominio del valor como del tiempo
2. Fallo de retraso, un sistema produce servicios correctos en el dominio del valor pero no en el del tiempo.
3. Fallo de silencio, cuando el sistema falla bruscamente sin haber tenido fallos de valor o de tiempo, a partir del fallo todos los servicios subsiguientes también sufren fallos de omisión.
4. Fallo de parada, un sistema que cumple los requisitos de un fallo de silencio pero permite que otros sistemas detectan que ha entrado en el estado de fallo
5. Fallo controlado, un sistema falla de una forma especificada y controlada.
6. Sin fallos, un sistema produce los servicios correctos, tanto en el dominio del valor como del tiempo.

## 3. Excepciones y manejo de excepciones

Se continúa con el estudio realizado en el tema anterior sobre la producción de componentes de software fiables, centrándose este tema en el manejo de las excepciones, estudiándose tanto los modelos de terminación como los de reanudación.

## **4. Programación concurrente**

En este tema se introduce la noción de proceso, tarea y hebra o hilo y revisa los modelos que utilizan los diseñadores de lenguajes y de sistemas operativos. El término tarea se utiliza genéricamente para representar una actividad concurrente. Se aprovecha también este tema para estudiar la distribución de tareas cuando se dispone de multiprocesadores o de sistemas distribuidos. Dejándose para los siguientes dos temas la comunicación entre tareas.

## **5. Sincronización y comunicación basada en variables compartidas**

Este tema, junto con el siguiente, se centra en el estudio de la comunicación entre tareas. En concreto en este primer tema se escriben los métodos de variables compartidas, incluyendo la utilización de semáforos, monitores, variables compartidas y objetos protegidos.

## **6. Sincronización y comunicación basada en mensajes**

Este tema es la continuación del anterior y resalta la importancia que tienen en los lenguajes modernos los métodos basados en mensajes para la comunicación y sincronización.

## **7. Acciones atómicas, Tareas Concurrentes y Fiabilidad**

En este tema se amplian las discusiones iniciales sobre tolerancia a fallos describiendo con cuánta fiabilidad puede ser programada la cooperación entre procesos. Para esta discusión es fundamental la noción de acción atómica y las técnicas de manejo de eventos asíncronos.

## **8. Control de recursos**

Como continuación del tema anterior, en este tema se tratan los procesos competitivos. Un tema importante en este caso es la distinción entre sincronización condicional y sincronización evitable en el modelo de concurrencia.

## **9. Capacidades de tiempo real**

Los requisitos temporales constituyen la característica diferenciadora de los sistemas de tiempo real. En este tema se presenta una visión de estos requisitos y de las funcionalidades del lenguaje y estrategias de implementación que se utilizan para satisfacerlos. Los sistemas de tiempo real estrictos tienen restricciones de tiempo que deben ser satisfechas, los sistemas no estrictos fallan a veces a la

hora de cumplir dichas restricciones adecuadamente. Los dos casos se consideran en el contexto de la planificación con tiempos límite.

## **10. Planificación**

Tras el tema anterior de capacidades de tiempo real, hay que incluir estas capacidades en la planificación de tareas, introduciéndose también la noción de prioridad junto con el análisis de la planificabilidad para sistemas con desalojo basado en prioridad.

## **11. Programación de bajo nivel**

Un requisito importante de muchos sistemas de tiempo real es que incorporan dispositivos externos que deben ser programados (controlados) como una parte del software de la aplicación. Esta programación a bajo nivel no concuerda con la aproximación abstracta para la producción de software que caracteriza a la ingeniería del software. En este tema se considera las formas en que las funcionalidades de bajo nivel pueden ser incorporadas con éxito en los lenguajes de alto nivel.