6-2-2020

Lisbeth Martínez Velázquez

ing.mecatrónica 8-at/m prof:CARLOS gARABITO



Interrupciones

Programación de sistemas de embebidos

**Contexto de la Investigación (Sistemas Embebidos y de Tiempo Real)**

Dos fuerzas significativas provocan que los sistemas de cómputo penetren cada vez más en la vida cotidiana de las nuevas generaciones de seres humanos: por un lado, está la necesidad cada vez más creciente en la sociedad moderna de equipos con mayores prestaciones, más sofisticados e “inteligentes” y por el otro, el rápido avance en el hardware, la miniaturización y la disminución de su costo. Esto trae como consecuencia que cada día la computadora se utilice en nuevas esferas de aplicación como un componente destinado a interactuar y controlar los sistemas y equipos en los cuales se incorpora. Esto ha llevado a la aparición de los denominados sistemas embebidos o empotrados.

**Un sistema embebido** es un sistema de cómputo incluido en otro sistema y que forma parte esencial de él. Dicho en otras palabras, no se percibe como un sistema de cómputo, sinó como el sistema de aplicación en sí. Por ejemplo, un teléfono celular o el inyector de gasolina de un automóvil. Los sistemas embebidos incorporan un conjunto de características significativas que los diferencian de los sistemas de escritorio. Entre ellas, una de las más importantes y que está presente en muchos de ellos es la necesidad de operar en tiempo real.

**Un sistema de tiempo real** es cualquier actividad de procesamiento de información o sistema de cómputo que tiene que responder con restricciones de tiempo y de manera predecible a eventos externos. Stankovic lo define como aquel sistema en el cual la corrección no sólo depende del resultado lógico del cómputo, sino también del tiempo en el cuál este se produce [88]. Es decir, cada resultado debe darse en un plazo de tiempo específico, no importa si es largo o pequeño, pero de incumplirse con estos plazos el resultado final se considera erróneo.

De manera general, las restricciones de tiempo en un sistema de tiempo real pueden ser arbitrariamente complicadas pero la más común es la existencia de plazos de cumplimiento que pueden clasificarse como duros (“hard”), firmes (“firm”), o suaves (“soft”). Un plazo se dice que es duro si la consecuencia de su incumplimiento puede ser catastrófica. Se dice que un plazo es firme si el resultado producido por la tarea correspondiente, deja de ser útil expirado el plazo, pero su incumplimiento no provoca consecuencias muy severas. Un plazo que no es ni duro ni firme se dice que es suave, esto es, el resultado tiene mayor utilidad si se produce a tiempo pero conserva un valor disminuido pasado el plazo de cumplimiento.

Una concepción errónea muy común y arraigada es la creencia de que un sistema de tiempo real sólo tiene que ser rápido. Básicamente, ser rápido generalmente es una condición necesaria, pero no suficiente. Un sistema de tiempo real necesita cumplir plazos explícitos y ser rápido en el caso promedio no garantiza este cumplimiento. En su lugar, la característica fundamental de un sistema de tiempo real es que su comportamiento debe ser predecible. Esto es, debiera ser posible demostrar en la fase de diseño que todas las restricciones de tiempo de la aplicación se cumplirán siempre que se cumplan ciertas condiciones (incluyendo suposiciones de fallos). Esto requiere el conocimiento previo de las cotas en los tiempos de ejecución y los plazos de todas las tareas de manera que puede ser analizado formalmente. De este modo, el diseñador puede tener una temprana advertencia de la inhabilidad del sistema de satisfacer sus requerimientos temporales y tomar así las acciones correctivas apropiadas. En otras palabras, **tiempo real no es sinónimo de rápido sino de predecible.**

**Sistema Operativo para Sistemas Embebidos y de Tiempo Real**

Los requerimientos antes descritos traen como consecuencia la necesidad de contar con sistemas operativos de propósito específicos sobre los cuales poder realizar este tipo de aplicaciones. A este tipo de sistemas operativos se les conoce como sistemas operativos de tiempo real.

La característica fundamental de un sistema operativo destinado a aplicaciones de tiempo real es la predecibilidad; es decir que suministre la capacidad de demostrar o probar que se satisfacen los requerimientos, sujeto a las suposiciones que se hagan. En especial, en los sistemas de tiempo real la predecibilidad se refiere a la posibilidad de demostrar el cumplimiento de los requerimientos temporales aún bajo suposiciones del peor caso. Esta cualidad está ausente en los sistemas operativos de propósito general. Un sistema operativo para aplicaciones de tiempo real confiables tiene que satisfacer los siguientes requerimientos:

* Soporta esquemas de planificación y sincronización que limitan el no determinismo propio de los sistemas concurrentes con el objetivo de garantizar las restricciones de tiempo (incluyendo plazos de las tareas).
* Soporta las necesidades de dominios de aplicación altamente confiables (por ejemplo soporte para detección de errores y condiciones excepcionales; así como la supervisión de plazos).
* Suministra un alto grado de control a los programas de usuarios. En un sistema operativo convencional las aplicaciones de usuario tienen un control muy limitado sobre funciones del sistema operativo tales como planificación, administración de memoria, control de dispositivos de E/S. En un sistema en tiempo real, sin embargo, resulta esencial permitir al usuario un control preciso sobre la prioridad y características temporales de las tareas, sobre el uso de paginación o intercambio de procesos y sobre qué procesos deben estar siempre residentes en la memoria principal.
* Suministra al programador un entorno simple y completamente predecible (conjunto restringido de servicios del OS con tiempos de respuesta deterministas).

Un aspecto importante a destacar es que un sistema operativo de tiempo real no es un sistema de tiempo real, un sistema operativo de tiempo real sólo permite el desarrollo de un sistema de tiempo real, tener tal sistema operativo no evita que sobre el se desarrolle un sistema que no satisfaga sus requerimientos de tiempo. Si por ejemplo se construye un sistema que necesita satisfacer restricciones de tiempo utilizando un sistema operativo de tiempo real pero que tiene que responder a comunicaciones a través de una red Ethernet, nunca será un sistema de tiempo real duro porque Ethernet como tal no es predecible. Por supuesto si se decide construir una aplicación encima de un sistema operativo con multitarea cooperativa como Windows 3.11, el sistema tampoco será de tiempo real ya que el comportamiento del sistema operativo es impredecible.

En otras palabras, un sistema de tiempo real contiene todos los elementos, incluyendo el hardware, el sistema operativo y el software específico de la aplicación. Un sistema operativo de tiempo real es solo un elemento del sistema completo de tiempo real. La característica fundamental de todos los elementos que lo integran es la predecibilidad.

**Bibliografía:**

[1] Luca Abeni, Giorgio Buttazo, “Support for Dynamic QoS in the HARTIK Kernel”, Proceeding of the IEEE Conference on Real-Time Computing Systems and Applications, Cheju Island, South Korea, 2000.

[2] Luca Abeni, Ashvin Goel, Charles Krasic, Jim Snow, Jonathan Walpole “A Measurement-Based Analysis of the Real-Time Performance of Linux”, Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS’02), 2002.

[3] Mike Accetta, Robert Baron, William Bolosky, David Golub, Richard Rashid, Avadis Tevanian, Michael Young, “Mach: A New Kernel Foundation for UNIX Development”, 93-113, USENIX Association Conference Proceedings, USENIX Association, June 1986.