

Sistemas en Tiempo Real

Mg. Hugo Ramón¹

Ing. Fernando Romero²

*Laboratorio de Investigación y Desarrollo en Informática³
Cátedra de Sistemas de Tiempo Real - Facultad de Informática - UNLP*

Resumen

El objetivo de esta línea de investigación es el estudio y desarrollo de aplicaciones que poseen restricciones temporales. Resulta de interés el estudio de diferentes metodologías para la especificación y desarrollo de aplicaciones en tiempo real, así como las plataformas donde dichas aplicaciones pueden ejecutarse.

Se trabaja experimentalmente con distintos tipos de hardware y software entre los que pueden mencionarse placas A/D, mini-robots, máquinas multimicroprocesadores y diferentes sistemas operativos.

Interesan especialmente las aplicaciones que involucren decisiones que van desde el hardware a utilizar hasta la plataforma sobre la que correrá dicha aplicación.

Palabras Claves

Tiempo Real, Ingeniería de Software, Dispositivos A/D, mini-Robots, RT-LINUX, QNX

Introducción

Los sistemas de tiempo real (STR) son sistemas de tiempo crítico, es decir, se trata de aplicaciones para las cuales el tiempo en que se produce la respuesta es un requerimiento esencial.

Por lo general, un STR incluye un conjunto de dispositivos independientes (hardware y software) que operan a diferentes velocidades.

¹ Profesor Adjunto Dedicación Exclusiva. Fac. de Informática. Universidad Nacional de La Plata. E-mail: hramon@lidi.info.unlp.edu.ar

² JTP Semidedicación. Fac. de Informática. Universidad Nacional de La Plata. E-mail: fromero@lidi.info.unlp.edu.ar

³ Calle 50 y 115 1er Piso, (1900) La Plata, Argentina, TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

En un STR se deben satisfacer requerimientos de performance. Comparando con un sistema tradicional, se debe tener especial control sobre la utilización de recursos, mejora en los tiempos de respuesta y manejo de distintas prioridades.

Existen algunas diferencias importantes entre un sistema de software tradicional y un STR:

- Control de dispositivos externos.
- Procesamiento de mensajes que pueden llegar en intervalos irregulares.
- Detección y control de condiciones de falla.
- Modelización de condiciones concurrentes.
- Alocaación y control de procesos concurrentes.
- Manejo las comunicaciones entre procesos.
- Protección de datos compartidos.
- Manejo de requerimientos de tiempos y performance.
- Testeo y debug de procesos concurrentes.
- Diseño de simuladores del hardware que no se encuentran disponibles durante la fase de pruebas.
- Selección del hardware adecuado para soportar el diseño del software.

Se espera que un STR se ejecute en forma continua, automática y segura, teniendo un impacto en los costos de desarrollo y la seguridad.

Esta caracterización indica claramente que un STR tiene que emplear herramientas de desarrollo y modelado diferentes a las utilizadas en el desarrollo de software tradicional.

Temas de Investigación y Desarrollo

Se plantean tres temas de estudios:

- Análisis de plataformas para desarrollo de aplicaciones de Tiempo Real
- Ingeniería de Software orientada a Tiempo Real
- Dispositivos de Tiempo Real

Análisis de plataformas para desarrollo en Tiempo Real

En este sentido se está estudiando el sistema operativo RT-Linux, explorando sus posibilidades para el desarrollo de aplicaciones distribuidas en tiempo real.

Se han realizado benchmark sobre sistemas operativos tradicionales para evaluar la posibilidad de ejecutar sistemas con restricciones blandas.

Ingeniería de Software orientada a Tiempo Real

Se estudian métodos de Ingeniería de Software orientados específicamente a las características del estudio del tiempo [GOLDSMITH 1993].

En este momento se incluyen herramientas de especificación como DTE, Redes de Petri y metodologías alternativas de diseño [GOMAA 1993].

Dispositivos para tiempo real

Se han desarrollado drivers para diversas placas conversoras a fin de que funcionen sobre diferentes plataformas.

Otros trabajos abarcan el uso de agentes físicos (mini-robots) móviles y fijos para los cuales se han desarrollado interfaces en diferentes plataformas.

Se han desarrollado simulaciones en OpenGL (GLScene) con aplicación docente y para testeado de aplicaciones utilizando como base experimental los mini-robots.

Líneas de Trabajo Futuras

Testing orientado a aplicaciones distribuidas en tiempo real [BEIZER 99].

Modelización con UML para sistemas complejos de tiempo real [DOUGLASS 2000].

Sistema operativo QNX, como plataforma de STR. Comparación con otros RTOS.

Definición de benchmark para evaluación de RTOS.

Bibliografía

[BEIZER 99] Software Testing Techniques. B. Beizer. International Thomson Computer Press, 1999.

[BUHR 1999] An Introduction to Real-Time Systems. L. Buhr. Prentice Hall, 1999.

[BURN 1996] Real-Time Systems and Programming Languages. A. Burns & A. Wellings. Addison Wesley, ISBN 90-201-40365-x

[DOUGLASS 2000] Real-Time UML: Developing Efficient Objects for Embedded Systems 2/ed. Addison Weley, 2000.

[ELLISON 1994] Developing Real-Time Embedded Software. S. Ellison, Wiley, 1994

[GOLDSMITH 1993] Practical Guide to Real-Time Systems Development. S. Goldsmith. Prentice Hall, 1993.

[GOMAA 1993] Software Design Methods for Concurrent and Real-Time Systems. H. Gomaa. SEI Series, 1993.

[GREHAN 1998] Real-Time Programming, Grehan. Addison Wesley, 1998.

[HATLEY 1988] Strategies for Real-Time System Specification, Hatley & Pirbahai, 1988, Dorset House, ISBN 0-932633-11-0

[LEVI 1990] Real Time System Design. Mc. Graw Hill, Levi & Agrawala, 1990, ISBN 0-07-037491-0

[NIELSEN 1990] ADA in Distributed Real-Time Systems, Nielsen, 1990, Mc. Graw Hill, ISBN 0-07-046544-4

[SHUMATE 1992] Software Specification and Design. A Disciplined Approach for Real-Time Systems, Shumate & Keller, 1992, Wiley, ISBN 0-471-53296-7

[SILBERSCHATZ 1994] Sistemas Operativos. Conceptos Fundamentales, 3rd Ed A. Silberschatz, J. Peterson, P. Galvin, Addison-Wesley Iberoamericana, 1994.

[SIMON 1999] An Embedded Software Primer, D. Simon, 1999. Addison Wesley CEPUB

Artículos IEEE & ACM