

Sistemas Distribuidos de Tiempo Real

Mg. Hugo Ramón¹ Ing. Fernando Romero²
Lic. Alejandro Gonzalez³ Lic. Pablo Novarini⁴ Sr. Cesar Estrebow⁵

Instituto de Investigación en Informática LIDI (III LIDI)⁶
Facultad de Informática - UNLP

Resumen

El objetivo de esta línea de investigación es el estudio y desarrollo de sistemas de software 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, plataformas de hardware (PC, PIC, etc.) y diferentes sistemas operativos (DOS, RT-Linux, QNX, etc.).

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

1 Profesor Dedicación Exclusiva. Fac. de Informática. UNLP, hramon@lidi.info.unlp.edu.ar

2 JTP Semidedicación. Fac. de Informática. UNLP, fromero@lidi.info.unlp.edu.ar

3 Ayudante Diplomado. Fac. de Informática. UNLP, agonzalez@lidi.info.unlp.edu.ar

4 Ayudante Diplomado. Fac. de Informática. UNLP, pnovarini@lidi.info.unlp.edu.ar

5 Ayudante Alumno. Fac. de Informática. UNLP, cesarest@lidi.info.unlp.edu.ar

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

Introducción

Los sistemas de tiempo real (STR) y los sistemas distribuidos de tiempo Real (SDTR) 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 o SDTR incluye un conjunto de dispositivos independientes (hardware y software) que operan a diferentes velocidades.

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 uno de tiempo real:

- 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 encuentra disponible durante la fase de pruebas.
- Selección del hardware adecuado para soportar el diseño del software.
- Distribución de tareas.

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

En particular nos interesan las aplicaciones industriales, caracterizadas por la distribución y el control del tiempo.

Calramente un STR tiene que emplear herramientas de desarrollo y modelado diferentes a las utilizadas en el desarrollo de software tradicional que es la temática correspondiente a los temas del Subproyecto 2 del III-LIDI en relación a la Ingeniería de Software de Sistemas Distribuidos y Sistemas Distribuidos en Tiempo Real.

Líneas de Investigación y Desarrollo

Se plantea 7 temas de estudio:

- Análisis de plataformas para desarrollo de aplicaciones de Tiempo Real

- Ingeniería de Software orientada a Tiempo Real
- Dispositivos de Tiempo Real
- Robótica
- Detección de trayectorias en ambientes controlados y abiertos

Resultados Obtenidos

- Se ha estudiado RT-Linux, explorando sus posibilidades para el desarrollo de aplicaciones distribuidas en tiempo real, realizando ejemplos concretos.
- Se han realizado benchmark sobre sistemas operativos tradicionales para evaluar la posibilidad de ejecutar sistemas con restricciones blandas.
- Se han realizado experiencia reales en RT-Linux y QNX: en cuanto al desarrollo de aplicaciones, análisis de sobrecarga y planificadores.
- Manejo de un robot empleando lógica difusa.
- Data logger utilizando como plataforma PIC MPS 430 (simulador)
- Seguimiento de Trayectoria

Trabajos Futuros

- Manejo de un robot empleando lógica difusa.
- Uso de simuladores de robots (Kephra).
- Aplicaciones de Robots a sistemas de producción industrial.
- Ampliación de capacidad de procesamiento del robot por empleo del PIC MPS 430.
- Utilización de MathLab en aplicaciones de control en Tiempo Real.

Formación de Recursos Humanos

- Se encuentran en desarrollo tres tesis de postgrado en temas relacionados a Elicitación de Requerimientos, commit y actualización de réplicas en bases de datos distribuidas y aplicaciones robóticas y una de grado en seguimiento y estimación de trayectorias.
- Utilización del miniRT y tinyRT como herramienta docente

Bibliografía

[BEIZER 1999] 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 Wesley, 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

[COULOURIS 1994] Distributed Systems Concepts and Design, Coulouris G., Dollimore J., Kindberg T., Addison Wesley 1994

IEEE Computer, IEEE Concurrency, IEEE Reliability and Maintainability Symposium.

IEEE Sigact News, IEEE Software Engineering, IEEE Transactions on Parallel and Distributed Processing

ACM Journal of Educational Resources in Computing, ACM Transactions on Computer Systems, ACM Transactions on Computer-Human Interaction, ACM Transactions on Database Systems, ACM Transactions on Information Systems, ACM Transactions on Software Engineering and Methodology.