

Planificación de Sistemas de Tiempo Real: Diseños y Problemas

Fernando Romero, Diego Encinas, Luciano Iglesias, Armando De Giusti¹,
Fernando G. Tinetti²

Instituto de Investigación en Informática LIDI (III-LIDI)
Facultad de Informática – Universidad Nacional de La Plata

{fromero, dencinas, li, degiusti, fernando}@lidi.info.unlp.edu.ar

RESUMEN

El objetivo de esta línea de investigación es el estudio y desarrollo de sistemas de software que poseen restricciones temporales, como son los Sistemas de Tiempo Real (STR) haciendo hincapié en los aspectos de simulación, planificadores y comunicaciones.

CONTEXTO

Esta línea de Investigación forma parte del proyecto "Arquitecturas multiprocesador distribuidas. Modelos, Software de Base y Aplicaciones" del Instituto de Investigación en Informática LIDI acreditado por la UNLP.

Palabras Claves: Tiempo Real, Sistemas Operativos de Tiempo Real, Sistemas Embebidos, Posix, comunicaciones, simulaciones, GNU/Linux.

1. INTRODUCCION

Un sistema de tiempo real (STR) [2] [12] [15] requiere reaccionar a estímulos del

medio ambiente, incluyendo el paso del tiempo físico, dentro de intervalos de tiempo dictados por dicho medio ambiente. Cada reacción o respuesta se lleva a cabo a través de un conjunto de instrucciones que normalmente se denominan tareas. Para asegurar que el sistema pueda ejecutar las tareas de tal manera que se cumplan con los plazos fijados por los intervalos de tiempos, se debe realizar una planificación rigurosa de tareas. Esta planificación es el punto central del diseño de aplicaciones de tiempo real. No puede quedar implícita simplemente por utilizar Sistemas Operativos de Tiempo Real (SOTR). Se debe conocer el funcionamiento de los planificadores implementados en estos SOTR y sus limitaciones. Es propósito de este proyecto de investigación abordar este tema, desde el punto de vista teórico y experimental. Para ello se dispone de simulaciones y sistemas reales. Debe tenerse en cuenta que los desarrollos teóricos no siempre acaban siendo implementados en sistemas reales. A su vez, los cambios en el hardware implican

¹ Investigador CONICET

² Investigador CIC Prov. de Buenos Aires.

nuevas posibilidades y variantes, tales como el hecho de disponer de varios núcleos de procesador en donde ejecutar tareas.

De las diversas implementaciones existentes de SOTR se estudia principalmente el sistema operativo GNU/Linux con parche RT, apto para tiempo real [22] [23] [25] [28]. Uno de los objetivos finales es la implementación de cambios en su planificador.

Las restricciones temporales pueden ser más o menos estrictas, por lo que en algunos sistemas se impone el uso de planificadores que definen y utilizan prioridades. Esta planificación se realiza sobre el recurso CPU. Si la tarea necesita hacer uso de otros recursos, estos no están planificados en los SOTR en general. Cuando uno de estos recursos es compartido por tareas de diversa prioridad, surge el problema de la inversión de prioridades, lo cual también es tema de este trabajo.

Comparando un STR con un sistema tradicional [24], advertimos se debe tener especial control sobre la utilización de recursos, tiempos de respuesta conocidos, manejo de prioridades, fallos y comunicaciones. Las principales diferencias entre estos sistemas son:

- Planificación de tareas dirigida al cumplimiento de restricciones de tiempo en vez de al rendimiento.
- Control de dispositivos externos, como por ejemplo, brazo robot, una planta industrial, etc., con un mayor uso de E/S.
- Procesamiento de mensajes que pueden arribar en intervalos regulares e irregulares.
- Seguridad y confiabilidad son conjuntas, debe haber detección y

control de condiciones de falla a fin de evitar daños y un funcionamiento sin interrupciones. Muchas veces estos sistemas no cuentan con supervisión humana, el manejo de excepciones es crítico. Se espera que un STR se ejecute en forma continua, automática y segura, teniendo impacto en los costos de desarrollo.

- Modelización de condiciones concurrentes, ya que al mezclarse eventos regulares e irregulares es frecuente su concurrencia y la de los procesos cuya ejecución desencadena dichos eventos.
- Manejo de las comunicaciones en Sistemas Distribuidos permitiendo reaccionar a los cambios de la red, como productividad (*throughput*) o retraso (*delay*).
- Protección de datos compartidos.
- Por lo general se trata de Sistemas Embebidos [13], forman parte de un entorno al cual controlan.

2. LINEAS DE INVESTIGACION Y DESARROLLO

Se plantean como temas de estudio:

- Verificación y validación del hardware, donde pueden encontrarse detalles o resultados de simulaciones que impliquen modificaciones. La simulación en software [9] [14] aclara este aspecto.
- Sistemas Operativos de Tiempo Real sobre microcontroladores: Free RTOS [10] y GNU/Linux con kernel RT en procesadores tipo ARM y x86.
- Planificación de CPU para/en STR [1] [4] [19] [20]: Simulaciones de algoritmos de planificación, utilizando herramienta Cheddar [6]. También se experimenta con la

planificación implementada en GNU/Linux RT sobre aplicaciones [17]. La finalidad es evaluar la conveniencia de implementar modificaciones en el planificador, tal como implementar la política EDF (Earliest Deadline First scheduling) [11] [16].

- Recursos compartidos: estudio de inversión de prioridades [3] [18] [21] [26] [27]. Se experimenta sobre SOTR, a fin de evaluar los mecanismos implementados en los mismos. Se implementan simulaciones para experimentar mecanismos no implementados, evaluándose la posibilidad de implementarlos en GNU/Linux RT.
- Se está simulando tráfico de red [5] [8] con restricciones temporales (streaming de audio y/o video), a fin de evaluar la conveniencia de los emisores.

3. RESULTADOS OBTENIDOS/ESPERADOS

De acuerdo con las tareas desarrolladas y a desarrollar, los resultados se enfocarán en varias direcciones relacionadas con sistemas operativos y con sistemas embebidos de tiempo real:

- Se estudian los sistemas operativos FreeRTOS y GNU/Linux RT.
- Identificación y definición de pautas para estrategias en Planificadores y su impacto en los sistemas en STR, ensayando las mismas en Cheddar. En particular, el planificador linux estandar Posix 1003 [7].
- Uso de robots y simuladores de robots Kephra. Odometría.

4. FORMACION DE RECURSOS HUMANOS

En base a estos temas se están desarrollando un Doctorado y dos tesis de Magister en Redes de Datos y posibles tesinas de grado. También aportan trabajos de alumnos de la materia Diseño de Sistemas de Tiempo Real.

5. BIBLIOGRAFIA

[1] N.C. Audsley , A. Burns , M. F. Richardson , A. J. Wellings. "Hard Real Time Scheduling: The Deadline Monotonic Approach". 1991. Proc. IEEE Workshop on Real-Time Operating Systems and Software.

[2] Alan Burns and Andy Wellings. "Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX", Addison-Wesley Educational Publishers Inc., 2009.

[3] Andreu Carminati, Romulo Silva de Oliveira, and Luis Fernando Friedrich. "Implementation and evaluation of the synchronization protocol immediate priority ceiling in PREEMPT-RT linux". Journal of Software, 7(3), March 2012.

[4] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. J. ACM, 20(1):46-61, 1973.

[5] Cisco visual networking index: Forecast and methodology, 2011-2016 [Visual networking index (VNI)]. Technical report.

[6] The Cheddar project: A free real time scheduling analyzer, <http://beru.univbrest.fr/~singhoff/cheddar/>

[7] David R. Butenhof, Programming with POSIX Threads. Addison-Wesley Professional, 1997.

- [8] R. Davis, «Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised.» Real-Time Systems. Springer, vol. 35, n° 3, pp. 239-272, 2007.
- [9] J. Eickhoff, Simulating Spacecraft Systems, Springer, 2009.
- [10] "FreeRTOS - market leading RTOS (real time operating system) for embedded systems supporting 34 microcontroller architectures". <http://www.freertos.org/>.
- [11] Giorgio C. Buttazzo University of Pravia Italy. Rate Monotonic vs EDF, Real-Time Systems, 29, 5–26, 2005 2005 Springer Science + Business
- [12] Giorgio C. Buttazzo, "Hard RealTime Computing Systems", Third edition, Springer, 2011.
- [13] Hermann Kopetz "Real-Time Systems, Design Principles for Distributed Embedded Applications". Second Edition. Springer. 2011.
- [14] IEEE Standards Association, IEEE Standard for standard SystemC language reference manual, IEEE Computer Society, 2012.
- [15] Jane W. S. Liu "Real Time Systems", Integre Technical Publishing Co., Inc., 2000
- [16] John A. Stankovic, "Deadline Scheduling for Real-Time Systems: Edf and Related Algorithms". Springer.1998.
- [17] Jinfeng Huang. "Predictability in Real-Time Software Design", Technische Universiteit Eindhoven. 2005.
- [18] L. Sha, R. Rajkumar, and J. P. Lehoczky. "Priority inheritance protocols: An approach to real-time synchronization". IEEE Trans. Comput., 39(9):1175-1185, September 1990.
- [19] Lui Sha, "Real Time Scheduling Theory: A Historical Perspective", University of Illinois at Urbana Champaign, Kluwer Academic Publishers. 2004.
- [20] Martin Hervé, Bothorel Tests de faisabilité basés sur le taux d'occupation CPU des principaux algorithmes d'ordonnancement temps réel.
- [21] "Mars pathfinder". <http://mars.jpl.nasa.gov/MPF/>.
- [22] Robert Love Linux Kernel Development Third Edition. Addison Wesley.2010. Media, Inc. Manufactured in The Netherlands.
- [23] "RTwiki". <http://rt.wiki.kernel.Org/index.php/Main Page>.
- [24] Silberschatz, J. Peterson, P. Galvin, "Sistemas Operativos. Conceptos Fundamentales", 3rd Ed. Addison-Wesley Iberoamericana, 1994.
- [25] Siro Arthur. Carsten Emde. Nicholas Mc Guire. "Assessment of the Realtime Preemption Patches (RT-Preempt) and their impact on the general purpose performance of the system". 20 nov 2007. 9th RT Workshop.
- [26] Tarek Helmy & Syed S. Jafri, "Avoidance of Priority Inversion in Real Time, Systems Based on Resource Restoration". International Journal of Computer Science & Applications, Vol. III, No. I, pp. 40-50, 2006
- [27] Victor Yodaiken."Priority inheritance in the kernel". <http://lwn.net/Articles/178253/>, 2002.
- [28] Wolfgang Mauerer. Professional Linux@Kernel Architecture. Wiley Publishing, Inc.2008.
- [29] William Stallings. High-speed networks and internets: performance and quality of service. Prentice Hall, 2002.