



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

**DISEÑO DE SISTEMAS DE TIEMPO
REAL**

Año 2013

Carrera/Plan:

Licenciatura en Informática
Plan 2003-07./ Plan 2012

Año: 5º

Régimen de Cursada Semestral

Carácter: Obligatoria

Correlativas: Ingeniería de Software II
– Laboratorio de Software

Profesor: *Fernando Romero*

Hs semanales: 6 hs

FUNDAMENTACIÓN

En esta materia se pone al alcance de los alumnos los conocimientos de hardware y herramientas de ingeniería y desarrollo de software necesario para la construcción de sistemas de tiempo real. El abordaje de los temas incluye partes teóricas complementadas con laboratorios prácticos, partiendo desde sistemas pequeños hasta grandes.

OBJETIVOS GENERALES:

Caracterizar los sistemas de tiempo real (STRs) y los sistemas distribuidos de tiempo real (SDTRs), en particular en relación con el desarrollo de software para los mismos. Estudiar aspectos propios de la arquitectura y comunicaciones de los SDTRs. Plantear las extensiones de la metodología clásica de Ingeniería de Software para SDTR y analizar herramientas para su análisis, diseño y verificación. Estudiar herramientas de especificación (Diagramas de Estados, Redes de Petri Extendidas) y aspectos de lenguajes orientados a tiempo real. Aplicar los conceptos teóricos en casos concretos de adquisición y control de datos en tiempo real, en particular orientados a aplicaciones industriales. Estudiar Sistemas Operativos orientados al desarrollo de aplicaciones distribuidas en tiempo real.

CONTENIDOS MINIMOS:



- Características de los Sistemas de Tiempo real y su software.
- Analizar los problemas asociados con la distribución de procesamiento y datos en STR.
- Ingeniería de Software de SDTR.
- Herramientas de especificación y lenguajes de programación para SDTR.
- Aplicaciones a control industrial y robótica.
- Sistemas operativos orientados a SDTR.

PROGRAMA ANALÍTICO

Unidad 1: Arquitectura de un Sistema de Tiempo Real (STR)

Caracterización de los Sistemas de Tiempo Real. Asincronismo.
Concurrencia. Interfaces de hardware. Drivers de dispositivos.
Manejadores de interrupciones. Concurrencia: procesos y su comunicación.
Caracterización de sistemas distribuidos.
Clasificación de los sistemas de Tiempo Real.
Taxonomía de las arquitecturas multicomputador y multiprocesador .
Arquitecturas multi-microprocesador.
Aplicaciones en tiempo real.
Generalización de modelos clásico de ciclo de vida, aplicado a problemas de tiempo real. Comunicación entre procesadores y procesos.
Topología de red. Buses.
Problemas clásicos: Arbitraje, detección de fallas, acceso al medio de comunicación. Modelos de buses.
Tolerancia fallas en arquitecturas distribuidas.

Unidad 2: Conceptos de Software para STR

Efecto de las características de los SDTR sobre el desarrollo de software.
Herramientas de especificación e implementación de STRs.
Verificación de STR.
Metodología de diseño. Hardware first-Software first.
Modelos de comunicación y sincronización de procesos.
Modelos de sincronización por Memoria Compartida.
Modelos de sincronización por mensajes asincrónicos.
RPC y Rendezvous.
Lenguajes y primitivas propias de los lenguajes asociados con cada modelo.

Unidad 3: Distribución de datos en los SDTR

Datos distribuidos.
Bases de datos distribuidas. Problemas asociados.
Consideraciones de diseño.



Replicación de datos. Redundancia y consistencia.
Máquinas dedicadas al Servicio de Base de Datos.
Soporte de Sistema operativo para sistemas de datos distribuidos que se acceden en tiempo real.

Unidad 4: Herramientas de especificación y desarrollo de SDTR

Caracterización y especificación de sistemas distribuidos de tiempo real.
Redes de Petri como herramientas de especificación STRs.
Análisis de STRs y SDTRs con Redes de Petri y Redes de Petri extendidas.
Análisis de restricciones de tiempo con RPE.
Otras herramientas de especificación formal. Relación con los lenguajes de programación.

Unidad 5: Ingeniería de Software para STR

Extensiones a la metodología estructurada para tratar problemas de Tiempo Real.
Herramientas de análisis y diseño.
Diagrama de Arquitectura.
Extensiones para el análisis y diseño teniendo en cuenta las restricciones de tiempo.
HRT-HOOD (Hard Real-Time Hierarchical Object-Oriented Design)
UML & STR.
Análisis de casos concretos.

Unidad 6: Aplicaciones

Aplicaciones en control de hardware.
Aplicaciones en robótica con y sin movimiento del robot. Aplicaciones en tratamiento de señales, en particular imágenes.

METODOLOGÍA DE ENSEÑANZA

Se trata de una materia con un enfoque teórico práctico pero con un peso mayor de trabajos prácticos de laboratorio. Dichas prácticas se pueden realizar en la Sala de pc de la Facultad en el horario de consultas de trabajos prácticos o en forma autónoma, por parte de los alumnos, ya que se pueden bajar desde la plataforma WEBUNLP a sus propias computadoras los simuladores para su realización.

En detalle se tratan de:

-Simulador utilizando BOSCH de arquitectura ia386 con una placa adquisidora A/D, más interfaz gráfica para simular entradas digitales y analógicas (desarrollado en la Cátedra).



-Live cd de linux-rt, con las herramientas instaladas necesarias (compiladores, herramientas de test, etc.) (los cd fueron armados por la Cátedra)
-Herramientas varias de software libre para desarrollo y simulación de microcontroladores Microchip (MPLAB, Proteus) y ARM (Keil, Proteus).

También se dispone en la Cátedra de 3 brazos robot tipo LynxArm, otro robot tipo vehículo (armados en la Cátedra), y 3 robots Khepera. Este equipamiento está disponible para los alumnos que quieran hacer un trabajo para promocionar la materia.

EVALUACIÓN

La aprobación de la cursada-trabajos prácticos y final de esta materia contempla dos modalidades:

-Modalidad Presencial:

Presentación obligatoria de los trabajos prácticos dentro de los plazos de la cursada, con corrección y nueva entrega, hasta que estén finalizados y funcionando. Luego se rinde un examen parcial sobre los temas vistos en dichas prácticas. **La aprobación del examen parcial práctico otorga la aprobación de TP.**

En esta modalidad se puede **promocionar la materia (habiendo aprobado los TP)**, realizando un trabajo adicional monográfico sobre un tema a designar por la Cátedra. Esta monografía se puede presentar en cualquier fecha de examen final a la que el alumno se anote, dentro del período de validez de los TP.

No habiendo cumplido con la aprobación de algún trabajo práctico o del parcial práctico, los alumnos pasan a la modalidad de aprobación a distancia.

-Modalidad a distancia:

Se realizará un trabajo teórico-práctico a determinar por la Cátedra, sobre alguno de los temas que trata la materia.

Los alumnos pueden consultar mediante el entorno de EAD Web-UNLP y además disponen de horarios presenciales para consulta, que se publicarán en el sitio WEB de la cátedra.

Se toma un parcial sobre los temas de las prácticas cuya aprobación otorga la aprobación de los TP.

El examen final consiste en la defensa del trabajo teórico-práctico resuelto por el alumno, anotándose en cualquier fecha de examen final durante la validez de los TP aprobados.

En ambas modalidades los alumnos disponen de la plataforma WEBUNLP para realizar consultas, o enviar los trabajos a presentar para su corrección. En dicha plataforma los alumnos disponen de las transparencias de teoría, las prácticas, el material para realizar las mismas, tanto simuladores como imágenes para armar live cd's con los sistemas operativos de tiempo real con los que trabajar. También un foro para hacer consultas a los auxiliares de la Cátedra.



BIBLIOGRAFÍA OBLIGATORIA

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

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

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

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

A practical Guide to Real-Time Systems Development. Prentice Hall, 1993.

Designing concurrent, distributed and real-time applications with UML. Addison Wesley, ISBN 0-201-65793-7.

BIBLIOGRAFÍA COMPLEMENTARIA

1. S. A. Aldarmi, A. Burns. Dynamic Value-Density For Scheduling Real-Time Systems . Noviembre 1998, Real Time System Group, University of York, Technical Report number YCS-98-310.

2. M. Aldea Rivas, J. Miranda, M. González Harbour. Integrating Application-Defined Scheduling with the New Dispatching Policies for Ada Tasks. Ada-Europe'2005. pp.220~235

3. N. Audsley. Real Time Scheduling . Predictable Dependable Systems, vol 2, chapter 2, Part II.1995.

4. A. Burchard and J. Liebeherr and Y. Oh and S. H. Son Assigning Real-Time Tasks to Homogeneous Multiprocessor Systems . Technical Report CS-94-01, University of Virginia, 1994 Department of Computer Science.

5. G. Buttazzo. Rate Monotonic vs. EDF: Judgment Day. Real-Time Systems, Vol. 29, Issue 1, pp. 5-26, Enero 2005.

6. AADL performance analysis with Cheddar : a summary. Dissaux P, Legrand J, Plantec A and Singhoff F. AADL SAE Spring working group meeting . Sevilla (Spain). Abril 2008.

7. S. Edzang. Simulation d'algorithmes d'ordonnancement temps réel. Rapport de travail d'étude et de recherche, Junio 2006.

8. <http://www.freertos.org/>

9. L. George, N. Rivierre and M. Spuri. Preemptive and Non-Preemptive Real Time Uni-Processor Scheduling.

INRIA Research Report number 2966. Septiembre 1996.

10. J. Goossens and C. Macq. Limitation of the Hyper-Period in Real-Time Periodic Task



Set Generation . 9th Conference RTS embedded systems 2001, Paris (France). Page 133-148.

11. K. Jeffay, D. F. Stanat, C. U. Martel. On Non-Preemptive Scheduling of Periodic and Sporadic Tasks . In the Proceedings of the RTSS91 Real Systems Symposium. San Antonio, Texas. Diciembre 1991.

12. G. Koren, D Shasha. D-Over, an optimal On-line scheduling algorithm for overloaded real time systems INRIA Technical report number RT-0138. Febrero 1992.

13. C. L. Liu and J. W. Layland Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment. 1973, Enero, 20(1):46-61, Journal of the Association for Computing Machinery.

14. Tests de faisabilité basés sur le taux d'occupation CPU des principaux algorithmes d'ordonnancement temps réel H. Martin, S. Bothorel. Rapport de travail d'étude et de recherche, Mars 2004.

15. Y. Oh and H. S. Son. Tight performance bounds of heuristics for a real-time scheduling problem. 1993.

16. Bo Su; Li Wang; , "Application of Proteus virtual system modelling (VSM) in teaching of microcon-troller," E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on , vol.2, no., pp.375-378, 17-18 April 2010

18. Sha, R. Rajkumar and J.P. Lehoczky. Priority Inheritance Protocols : An Approach to Real Time Synchronization. IEEE Transactions on computers, 39(9):1175-1185. 1990



CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Contenidos/Actividades	Evaluaciones previstas
1	Arquitectura de un Sistema de Tiempo Real. Introducción y conceptos. Comunicaciones. Manejadores de interrupciones Conversión analógico digital	Entrega de práctica de laboratorio 24/4
2	Sistemas operativos de tiempo Real: RTLinux, RTAI, LinuxRT	Entrega de práctica de laboratorio 22/5
3	Microcontroladores: Sistemas pequeños Microcontroladores: Sistemas con SOTR	Entrega de práctica de laboratorio 26/6
4	Planificación de procesos. Uso de Cheddar	
5	Bases de datos de tiempo real	
6	Ingeniería de software. Redes de Petri. HRT Hood	
7	Robots	
8	Sincronización de relojes	
		Evaluación parcial de los contenidos vistos 3/7 con recuperatorio 10/7

Contacto de la cátedra (mail, página, plataforma virtual de gestión de cursos):

<http://webunlp.unlp.edu.ar>



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

Firmas del/los profesores responsables: