



**DISEÑO DE SISTEMAS DE TIEMPO
REAL**

Año 2012

Carrera/Plan:

Licenciatura en Informática Plan 2003/07-2012.

Licenciatura en Sistemas Plan 2003/07-2012

Año: 5º

Régimen de Cursada Semestral

Carácter: Electiva (Lic. en Informatices)
Optativa (Lic. en Sistemas)

Correlativas:

Ingeniería de Software II

Laboratorio de Software

Profesor: Fernando Romero

Hs semanales: 6 hs

FUNDAMENTACIÓN

Los alumnos adquieren conceptos teóricos y práctica en el diseño de aplicaciones de tiempo real, desde sistemas simples en microcontroladores hasta complejos, tanto en Ingeniería de Software, elección de Hardware, Sistema Operativo, y modelo de programación..

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 asíncronos.

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.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

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

Clases teóricas alternadas con prácticas. En las teorías se utilizan medios audiovisuales, a fin de exponer los conceptos teóricos y ejemplificarlos. En las prácticas se utiliza una sala de PC, donde se encuentran instalados los sistemas operativos y herramientas tales como simuladores, compiladores. Se dispone de hardware especializado tal como placas conversoras analógicas digitales y robots, para las prácticas se han desarrollado simuladores, tanto de las placas como de los robots (virtualización de hardware), lo que permite que cualquier computador sea convertido en una máquina apta para la realización de la práctica. Lo mismo sucede con los Sistemas Operativos de Tiempo Real, de los cuales se dispone de versiones en Live CD, lo que evita tener que instalarlo en la máquina. Esta modalidad (virtualización y Live CD) permite que se puedan continuar trabajando fuera de las prácticas en sus PC o en las que la Facultad pone a su disposición en diferentes ámbitos.

EVALUACIÓN

La evaluación de la cursada consiste en la entrega de prácticas obligatorias grupales y un Examen parcial individual sobre las mismas prácticas. Para el seguimiento de los alumnos y consultas se utiliza una plataforma de enseñanza a distancia además de las clases y prácticas presenciales. En dicha plataforma se accede a los repositorios con los materiales necesarios para realizar las prácticas además de bibliografía en versión de apuntes de la cátedra y transparencias.

La aprobación final de la materia consiste en un examen final con contenidos teóricos y prácticos, existiendo la posibilidad de realizar como complemento a dicho examen un trabajo integrador de los diferentes conceptos vistos en la materia. De dichos trabajos a menudo surgen temáticas para trabajo de tesina de fin de carrera y publicaciones en congresos. En el caso del trabajo integrador, el examen final es un coloquio de defensa del trabajo.

BIBLIOGRAFÍA OBLIGATORIA

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

Schedulling in Real Time Systems. Francis Cottet, Joelle Delacroix, Claude Kaiser, Zoubir Mammeri 2002 John Wiley & Sons, Ltd. ISBN: 0-470-84766-2

BIBLIOGRAFÍA COMPLEMENTARIA



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

Barba-lace, A.; Luchetta, A.; Manduchi, G.; Moro, M.; Soppelsa, A.; Taliercio, C.: Performance Comparison of VxWorks, Linux, RTAI, and Xenomai in a Hard Real Time Application. Real-Time Conference, 15th IEEE-NPSS, (2007).

Dave Beal: RTAI (Real Time Application Interface). <http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/RTAI-RealTime-Application-Interface/> (2000)

L. Buhr. "An Introduction to Real Time Systems". Prentice Hall (1999).

FSM Labs, Inc.: Getting Started with RTLinux. http://www.ee.nmt.edu/~rison/ee352_fall09/Getting_Started_with_RTLinux.pdf (2001).

OSADL: Open Source Automation Development Lab <https://www.osadl.org/>

Giovanni Racciu, Paolo Mantegazza: RTAI 3.4 User Manual. [https://www.rtai.org/index.php?module=documents &JAS_DocumentManager_op=downloadFile&JAS_File_id=46](https://www.rtai.org/index.php?module=documents&JAS_DocumentManager_op=downloadFile&JAS_File_id=46) (2006)

Ismael Ripoll, Pavel Pisa, Luca Abeni, Paolo Gai, Agnes Lanusse, Sergio Saez: RTOS State of the Art Analysis. DISCA, Universidad Politecnica de Valencia, http://www.mnis.fr/ocera_support/rtos/ (2002).

Real-Time Linux: RT Preempt Kernel Patch https://rt.wiki.kernel.org/index.php/Main_Page

RTLinux Portal at Universidad Politécnica de Valencia. <http://rtportal.upv.es/>

Victor Yodaiken: An Introduction to RTLinux. <http://www.Linuxfordevices.com/c/a/Linux-For-Devices-Articles/An-Introduction-to-RTLinux/> (1999).

Diego Lopez Zamarrón: Análisis de Sistemas Operativos de Tiempo Real. <http://gayuba1.datsi.fi.upm.es/~dlopez/> (2004).



CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Contenidos/Actividades	Evaluaciones previstas
Teoría 1: Introducción y conceptos	Caracterización de SOTR	
	Planificación de SOTR	Practica en laboratorio
Teoría 2: Interrupciones y manejo de Tiempo	Elementos de hardware	
	Tiempo y SO	
	Sincronización	Practica en laboratorio
Teoría 3: datos en tiempo real	Magnitudes analogicas y digitales	
	Adquisicion de datos	
	Transductores	Practica en laboratorio
Teoría 4: S.Op.TRI	Diferencias con tradicionales	
	Planificadores	
	Inversión de prioridades	Practica en laboratorio
Teoría 5: Microcontroladores	Arquitecturas	
	Herramientas de desarrollo	
	Desarrollo de aplicaciones	Practica en laboratorio
Teoría 6: Ingeniería de software	Redes de Petri	
	HRT Hood	
	Cheddar: Simulación planificación	
		Evaluación parcial y final

Contacto de la cátedra (mail, página, plataforma virtual de gestión de cursos): fromero@lidi.info.unlp.edu.ar, webunlp.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Firmas del/los profesores responsables: