



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Sistemas de Tiempo Real

Carrera: Ingeniería en Computación
Profesor Responsable: Romero, Fernando
Año: 4º
Duración: Semestral
Carga Horaria Semanal: 6hs
Carga Horaria Total: 96hs

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 Mínimos

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

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.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

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.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

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.

BIBLIOGRAFIA

Título: Real -Time Systems.Design Principles for Distributed Embedded Applications.
2ªEdition

Autores: Hermann Kopetz

Editorial: Springer

Año de edición 2011

Título: Hard RealTime Computing Systems Third edition

Autores: Giorgio C. Buttazzo

Editorial: Springer

Año de edición: 2011

Título: Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX

Autores: Alan Burns and Andy Wellings

Editorial: Addison-Wesley Educational Publishers Inc.

Año de edición: 2009

Descripción de las Actividades Teóricas y Prácticas

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. Se trabaja con:



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

- Placas de microcontrolador ARM, kit Sinectis Tower K70.
- Placas EDU CIAA.
- Placas Intel-Galileo.
- Kit Arduino con sensores.
- 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).
- Simulador basado en QEMU que implementa una virtualización de un microcontrolador ARM con Linux RT-Preempt com sistema operativo.
- Herramienta Cheddar, que permite simular procesos/threads bajo diferentes tipos de planificadores a fin de observar la factibilidad de su planificación.
- V-REP: simulador de robótica, permite simular diversos tipos de robot, incluido brazos de robot como los que dispone la Cátedra, y robots Khepera ii, también disponibles. También se dispone en la Cátedra de 3 brazos robot tipo LynxArm, otro robot tipo vehículo (armados en la Cátedra), y 3 robots Khepera ii. Este equipamiento está disponible para los alumnos que deseen realizar un trabajo para promocionar la materia.

Metodología de Enseñanza y Evaluación

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

-Modalidad con final:

- Presentación obligatoria de los trabajos prácticos dentro de los plazos de la cursada, con corrección y nueva entrega, hasta que estén finalizados y funcionando.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

- Luego se rinde un examen parcial sobre los temas vistos en dichas prácticas.
La aprobación del examen parcial práctico otorga la aprobación de TP.
- Examen final, que puede ser reemplazado por un trabajo con una parte monográfica y otra experimental sobre alguno de los temas vistos en la materia. La aprobación en este caso se lleva a cabo previa exposición del trabajo y breve coloquio sobre temas generales.

-Modalidad promoción:

- 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 sobre un tema a designar por la Cátedra. Este trabajo se debe realizar en forma conjunta con la cursada y debe presentarse el informe antes del cierre de las cursadas en febrero (esta materia es del segundo semestre). Pasado este plazo se ingresa a la modalidad con final.