

Asignatura: Programación en Tiempo Real

Carrera: Ingeniería en Informática

Ciclo Lectivo: 2016

Docentes: Coordinador: Mg. Ing. Diego Omar Encinas

Carga horaria semanal: 5 horas

Tipo de Asignatura: Teórico-práctica.

Fundamentación: Programación en Tiempo Real es una materia obligatoria correspondiente al cuarto año de la carrera de Ingeniería en Informática.

Se utilizan los conceptos desarrollados en la materia Complejidad Temporal, Estructura de Datos y Algoritmos junto con los estudiados en el transcurso de las unidades. En la materia los alumnos abordarán los conocimientos/herramientas de software y de hardware sobre programación para sistemas de tiempo real.

Objetivos: Al finalizar el curso, el estudiante será capaz de:

Comprender y manejar conceptos de Sistemas de Tiempo Real.

Desarrollar y caracterizar los sistemas de tiempo real y los sistemas distribuidos de tiempo real

Aplicar los conceptos teóricos en casos concretos de adquisición y control de datos en tiempo real, en particular orientados a aplicaciones industriales.

Plantear las extensiones de la metodología clásica de Ingeniería de Software para SDTR y analizar herramientas para su análisis, diseño y verificación.

Utilizar herramientas de desarrollo e implementación (simulación, lenguajes de descripción de sistemas).

Contenidos mínimos: Sistemas de Tiempo Real (STRs). Arquitectura de un STR. Herramientas de Análisis, Verificación y Diseño. Ingeniería de Software para STRs. Concurrencia. Sincronización. Aplicaciones. Control de Hardware. Drivers. Tratamiento de Señales. Sistemas Operativos orientados a aplicaciones en tiempo real.

Unidades temáticas:

Unidad I

Conceptos Básicos de Programación en C. Tipos de Datos. Estructuras de Datos. Estructuras de Control. Administración de Memoria. Manejadores de interrupciones. Metodología de diseño. Hardware first-Software first.

Unidad II

Definición de Sistemas de Tiempo Real. Arquitectura de un Sistema de Tiempo Real. Herramientas de Análisis, Verificación y Diseño. UML & STRs. HRT-HOOD (Hard Real-Time Hierarchical Object-Oriented Design). Taxonomía de las arquitecturas multicomputador, multiprocesador y multi-microprocesador.

Unidad III

Programación Concurrente. Procesos. Sincronización. Exclusion Mutua. Semáforos. Memoria Compartida. Comunicación entre procesadores y procesos. Modelos de sincronización por mensajes asincrónicos. RPC y Rendezvous.

Unidad IV

Aplicaciones. Arbitraje, detección de fallas, acceso al medio de comunicación. Modelos de buses. Tolerancia a fallas en arquitecturas distribuidas. Control de Hardware. Drivers. Tratamiento de Señales.

Bibliografía:

- Karl-Erik Arzen, Real-Time Control Systems, Department of Automatic Control, Lund Institute of Technology, Lund 2001.
- A. Burns and A. Welling, Real-Time Systems and Programming Languages, Addison Wesley, 2009.
- E. Bianchi, L. Dozio, P. Mantegazza., "RTAI, A Hard Real Time support for LINUX", Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano, 2001.
- Giorgio C. Buttazzo, "Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications" Springer; 2nd ed. Edition.
- Designing concurrent, distributed and real-time applications with UML. Addison Wesley, ISBN 0-201-65793-7.

Modalidad de dictado: Las clases se desarrollarán en el Laboratorio de Informática. Se organizarán en modalidades teórico-prácticas con soporte de presentaciones digitales y prácticas en función de cada clase.

En las clases se presentan los contenidos teóricos y se van resolviendo en forma conjunta ejemplos que ayuden a comprender los nuevos conceptos introducidos.

La formación práctica está basada en la resolución de problemas tipo, de actividades experimentales y de problemas abiertos de ingeniería.

En cuanto a la formación experimental se realizarán las siguientes actividades:

- Introducir al alumno al análisis de la problemática de entrada/salida: adquisición de datos, entradas y salidas digitales y analógicas por medio de simulación de arquitecturas (Simulador BOSCH).
- Introducir al alumno al uso de sistemas operativos de tiempo real por medio de Live cds de linux-rt, con las herramientas instaladas necesarias (compiladores, herramientas de test, etc.).
- Introducir al alumno al diseño de sistemas de tiempo real por medio de herramientas varias de software libre para desarrollo y simulación de microcontroladores Microchip (MPLAB, Proteus) y ARM (Keil, Proteus).

La realización de estas actividades les garantiza a los estudiantes una adecuada formación experimental, a partir de la aplicación de los conocimientos adquiridos hasta entonces.

En cuanto a los problemas abiertos de ingeniería, se realizarán trabajos relacionados a la implementación de una aplicación determinada, que haga uso de las herramientas de diseño de algoritmos y sistemas de programación de tiempo real, integrando los conceptos de las unidades temáticas desarrolladas durante la cursada y los estudiados en la asignatura Complejidad Temporal, Estructura de Datos y Algoritmos. La realización de los trabajos implica la identificación de un problema dado y la solución del mismo, a partir de la aplicación de los conocimientos adquiridos hasta entonces, lo cual constituye la base formativa para que el estudiante adquiera las habilidades que le permitan encarar proyectos y diseños de ingeniería. Cada trabajo debe incluir un detalle de los problemas encontrados, las formas de solucionarlos, las condiciones de ejecución, formato de los datos de entrada e ideas o sugerencias para realizar una versión mejorada del mismo.

Los estudiantes deben realizar entregas de al menos un trabajo integrador durante el desarrollo de la cursada el cual será revisado por los docentes y devuelto a los estudiantes para su corrección.

El material correspondiente a las clases teóricas, así como los documentos de la práctica se encontrarán disponibles a través de la plataforma UNAJ Virtual. Este mecanismo también será utilizado para realizar consultas simples.

Actividades extra-áulicas: Se establecerán guías de actividades prácticas para que el estudiante pueda ejercitar, a fin de consolidar los conceptos aprendidos en clase.

Evaluación: La asignatura puede aprobarse mediante régimen de promoción directa (sin examen final). La evaluación integradora de las instancias teórico-prácticas se realiza a través de dos parciales teórico-práctico en máquina. Además, los estudiantes deberán desarrollar un trabajo de investigación final donde se integren los temas vistos en la materia. Las clases son obligatorias ya que implican participación y debate que forman parte de la evaluación, el porcentaje de asistencia requerido es del 75%.

Se tomarán dos parciales con sus respectivos recuperatorios.

Aprobación de la materia:

Promoción con nota final mayor o igual a 7 y con notas parciales mayores o iguales a 6.

Cursada aprobada para nota de 4 a 6,99.

Desaprobado para nota de 0 a 3,99.

Al finalizar la cursada cada estudiante tendrá una calificación de las evaluaciones (NE) obtenida del promedio de los dos parciales aprobados y una calificación del trabajo de investigación (NTI).

La nota final de la asignatura se obtiene de la siguiente manera:

$NOTA\ FINAL = 0,8 * NE + 0,2 * NTI$