

Programa de Sistemas Embebidos Avanzados II



Código: EL

Identificación y características de la Actividad Curricular

Carrera/s:	Ingeniería Electrónica	Carácter:	Electiva
Plan de Estudios:	2014	Área:	Informática
Bloque/Campo:	Tecnologías Aplicadas	Formato Curricular:	ECE
Régimen de cursado:	Cuatrimstral	Departamento:	Sistemas e Informática
Cuatrimstre:	10º [ECA]		
Carga horaria:	64 hs / 4 hs semanales		
Escuela:	Ingeniería Electrónica		
Docente responsable:	Ing. José Luis Simón		

Programa Sintético

Estructura de los sistemas operativos de propósito general. Caso de estudio: Linux
Diseño y construcción de sistemas de tiempo real. Caso de estudio: FreeRTOS
Concurrencia y sincronización
Tratamiento de errores y excepciones
Conceptos de planificación
Modelado de aplicaciones en sistemas embebidos. UML

Asignaturas Relacionadas

Previas: Informática Aplicada, Sistemas Embebidos Avanzados

Simultáneas Recomendadas: **No Completar**

Posteriores: **No completar**

Vigencia desde 2020

.....
Firma Profesor

.....
Fecha

.....
Firma Aprob. Escuela

.....
Fecha

Con el Aval del Consejo Asesor:

Características Generales

Sistemas Embebidos II es una asignatura electiva del 10º cuatrimestre de la carrera de Ingeniería Electrónica de la FCEIA dictada en el marco establecido por la Escuela de Ingeniería Electrónica para las asignaturas electivas, cuyo propósito es ofrecer al futuro Ingeniero Electrónico un panorama de la problemática del desarrollo de software en los sistemas embebidos, los sistemas de tiempo real y la concurrencia en campos de aplicación críticos. Cuenta con un equipo de cuatro docentes, y se desarrolla en un conjunto de actividades de formación teórico-prácticas orientadas a la aplicación concreta de los conocimientos adquiridos.

Objetivos

- Presentar un panorama de la problemática de diseño y construcción de software desde la óptica de la Ingeniería Electrónica, con especial énfasis en los sistemas embebidos.
- Comprender la estructura y el funcionamiento de sistemas operativos utilizados en los sistemas embebidos y los conceptos de concurrencia, tiempo real, tratamiento de errores y planificación.
- Analizar y comprender el material relevante.
- Conocer los modelos de diseño de aplicaciones en sistemas embebidos.
- Conocer las tecnologías de diseño de software concurrente y de tiempo real.

Contenido Temático

UNIDAD I: SISTEMAS OPERATIVOS DE PROPÓSITO GENERAL - LINUX

- 1.1 Conceptos iniciales. Usuarios, permisos y roles. Línea de comando, uso del shell de Linux
- 1.2 Estructura de Linux: núcleo, HAL, API, manejadores de dispositivos
- 1.3 El sistema de archivos: tipos, conceptos, montaje, permisos. `chown` y `chmod`
- 1.4 Multiprocesamiento en Linux: procesos y threads. `fork()`, `ps`, `mpstat`.
- 1.5 Planificación en Linux: *scheduler*
- 1.6 Comunicación entre procesos: `signal`, `pipes` y `named pipes`
- 1.7 Sockets

UNIDAD II: INTRODUCCIÓN A LOS SISTEMAS DE TIEMPO REAL - FREERTOS

- 2.1 Sistemas de tiempo real: conceptos y caracterización
- 2.2 Sistemas operativos de tiempo real: el caso de estudio FreeRTOS
- 2.3 Requerimientos temporales: especificación
- 2.4 Criticidad y diseño orientado a eventos
- 2.5 Concurrencia en los STR: tareas, competencia y planificación. El API de concurrencia y sincronización de FreeRTOS
- 2.6 Modos de activación de las tareas
- 2.7 Comportamiento determinístico y estocástico en los STR
- 2.8 Diseño y desarrollo de software en los STR

UNIDAD III: CONCURRENCIA

- 3.1 Concepto de proceso, tarea e hilo. Propiedades y recursos
- 3.2 Comunicación entre procesos (IPC). Recursos y estrategias
- 3.3 Sincronización

3.4 La librería `pthread`. Conceptos y ejemplos de uso

UNIDAD IV: TRATAMIENTO DE ERRORES

4

.1 Conceptos fundamentales: fiabilidad, averías, errores y fallas

4

.2 Taxonomía de averías y fallas

4

.3 Mejoras en la fiabilidad: prevención y tolerancia a fallas

4

.4 Redundancia de software estática: programación de n versiones

4

.5 Redundancia de software dinámica. Fases

4

.6 Estrategia de bloques de recuperación

4

.7 Redundancia dinámica y excepciones

4

.8 Componente ideal de un sistema tolerante a fallas

UNIDAD V:

5

.1 Excepciones

5

.2 Representación de excepciones, dominio de un manejador de excepciones, propagación de excepciones, últimas voluntades, modelo de reanudación y de terminación

5

.3 Bloques de recuperación y excepciones

5

.4 Manejo de excepciones en C++: patrón RAII. Punteros inteligentes

UNIDAD VI: Modelos de desarrollo de aplicaciones en sistemas embebidos

6.1 Caracterización de las aplicaciones. Modelo de capas. Interfaces y APIs

6.2 Conceptos de distribución: canales, puntos de acceso y protocolos de transporte y aplicación.

6.3 Arquitectura de aplicaciones: modelos "cliente-servidor", "productor-consumidor".

6.4 Brokers y proxies

6.5 Patrones arquitecturales para el desarrollo y despliegue de aplicaciones distribuidas. Introducción a UML.

6.6 Implementación práctica de protocolos de aplicación: REST y OPC-UA

6.7 Caso de estudio: diseño de una aplicación DSP

Esta asignatura está basada en los lineamientos metodológicos definidos en el plan de estudios 2014. En particular, se cita: *“El proceso de formación del ingeniero priorizará el desarrollo de la creatividad, el diseño de innovaciones tecnológicas y la resolución de situaciones problemáticas con similitudes de realidad. Las estrategias de aprendizaje comprenderán procesos de integración de conocimientos, para lo cual, los trabajos por proyectos, simulación de realidad y resolución de problemas abiertos, entre otras, constituyen metodologías adecuadas”.*

Más del 50% de la carga horaria disponible para el desarrollo de los contenidos temáticos de la asignatura está destinado a la realización de proyectos que implican la resolución de problemas conceptuales y de aplicación en la ingeniería electrónica, en el campo específico del software.

Asimismo, los contenidos teóricos básicos y conceptuales, una vez desarrollados, se ejemplifican exponiendo su impacto en la implementación tecnológica final, contribuyendo también al volumen de conocimientos de índole aplicada impartidos en la materia.

De esta manera se intenta transmitir al alumno el conocimiento de las problemáticas tecnológicas mediante un enfoque orientado a la resolución y la aplicación, ya que el carácter de los contenidos implica en gran parte el desarrollo de habilidades que sólo se logran mediante el hacer.

Todas las actividades de índole práctico se realizan en el marco de proyectos de alcance acotado y ejecución grupal, para lo cual los alumnos tienen disponibles computadoras y kits de desarrollo basados en microcontroladores .

A los tiempos previstos para la carga horaria destinada a la materia dentro del plan, deben agregarse la disponibilidad de no menos de 6 horas semanales de consulta que los docentes en conjunto brindan a los estudiantes fueran del horario regular de cursado y que están destinadas a la clarificación de conceptos o explicaciones puntuales conforme la necesidad de cada alumno en particular.

Actividades de Formación Práctica

Durante el cursado los alumnos desarrollarán en modalidad grupal proyectos propuestos por la cátedra con las temáticas desarrolladas en clase. Cada proyecto debe aprobarse en una evaluación presencial.

Para el desarrollo y la ejecución de los proyectos los alumnos contarán con el apoyo del equipo docente de la cátedra, tanto durante las horas de dictado como en horas de consulta.

Evaluación

La promoción de la asignatura se obtiene cumpliendo los requerimientos enunciados a continuación:

1. Ejecución de los proyectos mediante entrega y aprobación del informe escrito.
2. Aprobación de un examen integrador

Nota:

La cátedra ofrece una instancia remedial para este último examen.

Distribución de la carga horaria

Presenciales

Teoría		30	hs
Práctica	Experimental de laboratorio		hs
	Experimental de Campo		hs

Resolución de Problemas y Ejercicios	10	hs
Problemas abiertos de ingeniería		hs
Actividades de Proyecto y Diseño	24	hs
Práctica Profesional Supervisada		hs
Total	64	

hs

Evaluaciones	6	hs
Dedicadas por el alumno fuera de clase	40	hs
Preparación Teórica	20	hs
Preparación Práctica	12	hs
Elaboración y redacción de informes, trabajos, presentaciones, etc.	8	hs

Bibliografía básica

Realtime embedded systems: Open-Source Operating Systems Perspective, Ivan Cibrario Bertolotti Gabriele Manduchi. CRC Press 2012

Sistemas de Tiempo Real y Lenguajes de Programación (3ª Edición).
Alan BURNS y Andy WELLINGS.
ADDISON-WESLEY Iberoamericana España
ISBN: 8478290583

Sistemas Operativos Modernos
TANENBAUM, ANDREW S.
Pearson Vue
ISBN: 9702603153

Programming with POSIX threads
David R. Butenhof.
(Addison-Wesley professional computing series)
ISBN 0-201-63392-2

Concurrent and Real-Time Programming in Java, Andy Wellings ISBN:047084437X
John Wiley & Sons

Pullum, Laura. (2001). Software Fault Tolerance Techniques and Implementation (Artech House Computing Library)

E. Dubrova, Fault-Tolerant Design: An Introduction, Kluwer Academic Publisher (2005)

Buttazzo, Giorgio. Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications – Springer, Third Edition (2011)

Buttazzo, Giorgio & Lipari, Giuseppe & Abeni, L. & Caccamo, Marco. Soft real-time systems: Predictability vs. efficiency, Springer (2005)

Presentaciones de la cátedra:
I: Plataformas
II: Sistemas de Tiempo Real
III: Concurrencia, partes I, II y III
IV: Fiabilidad y tolerancia a fallos
V: Excepciones
VI: Planificación
VII: Modelos de diseño de aplicaciones

Bibliografía complementaria

Fundamentos de Sistemas Operativos, 7a edición, Galvin, Silberschatz y Gagne
Practical guide to bare-metal C++, Alex Robenko, Creative Commons
Ingeniería de Software, Ian Sommerville
Sistemas Distribuidos, 3° ed.; Colouris, Dollimore, Kindberg. , Addison-Wesley

Recursos web y otros recursos

Página web de la asignatura, donde se consignan los enlaces y material relevante de la web
<http://www.dsi.fceia.unr.edu.ar>

Cronograma de actividades

Semana	Unidad	Tema	Actividad
1	I	Presentación - Plataformas	Clases de teoría
2	I	Modelos de desarrollo	Clase de teoría/Clase práctica
3	II	Sistemas de Tiempo Real	Clases de teoría
4	II	Sistemas de Tiempo Real	Clase de teoría/Clase práctica
5	III	Concurrencia	Clase de teoría/Clase práctica
6	III	Concurrencia	Clase de teoría/Clase práctica
7	III	Concurrencia	Clase de teoría/Clase práctica
8	IV	Tolerancia a fallas	Clases de teoría
9	IV	Excepciones	Clase de teoría/Clase práctica
10	V	Planificación	Clase de teoría/Clase práctica
11	V	Planificación	Clase de teoría/Clase práctica
12	VI	Aplicaciones distribuidas	Clases de teoría
13	VI	Aplicaciones distribuidas	Clases de práctica
14	TP	Presentación de proyectos	Exposición de proyectos grupos
15	TP	Presentación de proyectos	Exposición de proyectos grupos
16	EV	Examen Integrador	Evaluación