Facultad de Ciencias

GUÍA DOCENTE DE LA ASIGNATURA

G653 - Programación Paralela, Concurrente y de Tiempo Real

Grado en Ingeniería Informática
Obligatoria. Curso 3

Curso Académico 2023-2024

1. DATOS IDENTIFICATIVOS

Título/s	Grado en Ingeniería Informática			Tipología y Curso	Obligatoria. Curso 3
Centro	Facultad de Ciencias				
Módulo / materia	MATERIA PROGRAMACIÓN DE COMPUTADORES MÓDULO OBLIGATORIO				
Código y denominación	G653 - Programación Paralela, Concurrente y de Tiempo Real				
Créditos ECTS	6	Cuatrimestre		Cuatrimestral (1)	
Web	http://www.istr.unican.es/asignaturas/ppctr/				
Idioma de impartición	Español	English friendly	No	Forma de impartición	Presencial

Departamento	DPTO. INGENIERÍA INFORMÁTICA Y ELECTRÓNICA
Profesor responsable	JOSE JAVIER GUTIERREZ GARCIA
E-mail	josejavier.gutierrez@unican.es
Número despacho	Facultad de Ciencias. Planta: + 3. DESPACHO DE PROFESORES (3061)
Otros profesores	BORJA PEREZ PAVON MARIO IBAÑEZ BOLADO

2. CONOCIMIENTOS PREVIOS

Conceptos:
- Nociones básicas de programación
- Programación básica en Java y C

Asignaturas cursadas:
- Organización de computadores
- Introducción al software
- Métodos de programación
- Sistemas operativos

3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS DEL PLAN DE ESTUDIOS TRABAJADAS
Competencias Genéricas
(Aprendizaje) Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
Capacidad de análisis, síntesis y evaluación.
Capacidad de resolución de problemas aplicando técnicas de ingeniería.
Capacidad para argumentar y justificar lógicamente las decisiones tomadas y las opiniones.
Capacidad de trabajo en equipo.
Aprendizaje autónomo.
Creatividad.
Capacidad de iniciativa y espíritu emprendedor.
Competencias Específicas
Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.
Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.
Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

3.1 RESULTADOS DE APRENDIZAJE

- Conocer los fundamentos de la programación de computadores y del razonamiento sobre los programas en lo que respecta a la programación paralela.

  Conocer con familiaridad los riesgos y dificultades adicionales de la programación paralela y ser capaz de enfrentarlos con las técnicas apropiadas.

  Capacitar al alumno para concebir, especificar, diseñar, implementar y verificar aplicaciones informáticas en la que se utilice programación concurrente, proporcionando los conceptos básicos de concurrencia, sus ventajas, los problemas y las patologías que conlleva, así como los recursos y las primitivas de sincronización que históricamente se han introducido para realizar una programación concurrente segura.

  Capacitar al alumno para el desarrollo de programas concurrentes mediante el uso de lenguajes de programación que soportan concurrencia o haciendo uso de los servicios de un sistema operativo.

  Capacitar al alumno para que sea capaz de diseñar y analizar sistemas de tiempo real monoprocesadores, en los que los aspectos temporales son fundamentales para el correcto funcionamiento de la aplicación.

  Capacitar al alumno para el desarrollo de programas sencillos de tiempo real para monoprocesadores mediante el uso de lenguajes de programación o de los servicios de un sistema operativo.

4. OBJETIVOS

Formar al alumno en los conceptos básicos de la programación paralela. Se introducirán conceptos de metodología de la programación paralela que difieren sensiblemente de los métodos empleados en la programación secuencial y orientada a objetos. Se presentarán una serie de esquemas algorítmicos paralelos, que son ampliamente utilizados en diferentes aplicaciones. En cuanto al modelo de programación nos centraremos en el paradigma de memoria compartida. Para ello se desarrollarán las prácticas de programación utilizando el estándar OpenMP, así como los mecanismos proporcionados por el estándar de C++.

Formar al alumno en el diseño, especificación, implementación y verificación de aplicaciones informáticas en la que se utilice programación concurrente. En particular se introducirán los conceptos básicos de concurrencia y sincronización, los métodos formales para especificar y verificar los programas concurrentes, y se practicará la programación concurrente en Java y en C sobre POSIX, identificando sus ventajas y problemas en contraste con la programación secuencial.

Formar al alumno en el diseño, análisis e implementación de sistemas de tiempo real monoprocesadores sencillos. Para ello se introducirán los diferentes conceptos que se han considerado históricamente relevantes para la correcta planificación de los sistemas de tiempo real. También se experimentarán los mecanismos de tiempo real con C sobre POSIX.

5. MODALIDADES ORGANIZATIVAS Y MÉTODOS DOCENTES

ACTIVIDADES	HORAS DE LA ASIGNATURA
ACTIVIDADES PRESENCIALES	
HORAS DE CLASE (A)	
- Teoría (TE)	30
- Prácticas en Aula (PA)	
- Prácticas de Laboratorio Experimental(PLE)	30
- Prácticas de Laboratorio en Ordenador (PLO)	
- Prácticas Clínicas (CL)	
Subtotal horas de clase	60
ACTIVIDADES DE SEGUIMIENTO (B)	
- Tutorías (TU)	7,5
- Evaluación (EV)	7,5
Subtotal actividades de seguimiento	15
Total actividades presenciales (A+B)	75
ACTIVIDADES NO PRESENCIALES	
Trabajo en grupo (TG)	15
Trabajo autónomo (TA)	60
Tutorías No Presenciales (TU-NP)	
Evaluación No Presencial (EV-NP)	
Total actividades no presenciales	75
HORAS TOTALES	150

6. ORGANIZACIÓN DOCENTE

	CONTENIDOS	TE	PA	PLE	PLO	CL	TU	EV	TG	TA	TU-NP	EV-NP	Semana
1	BLOQUE I: PROGRAMACIÓN PARALELA Tema 1: Introducción a la Programación Paralela Tema 2: Paralelismo en C++ Tema 3: Análisis de rendimiento de programas paralelos Tema 4. Programación en Memoria Compartida: OpenMP Tema 5: Optimizar el Rendimiento de Programas OpenMP Prácticas de OpenMP y C++.	10,00	0,00	10,00	0,00	0,00	2,50	2,50	5,00	20,00	0,00	0,00	11-15
2	BLOQUE II: PROGRAMACIÓN CONCURRENTE Y DE TIEMPO REAL 1.- Concurrencia - Introducción a la programación concurrente - Sincronización - Concurrencia y sincronización en lenguajes de programación y sistemas operativos - Patrones y métodos de expresión de la concurrencia	10,00	0,00	10,00	0,00	0,00	2,50	2,50	5,00	20,00	0,00	0,00	1-5
3	BLOQUE II: PROGRAMACIÓN CONCURRENTE Y DE TIEMPO REAL 2.- Tiempo real - Introducción a los sistemas de tiempo real: políticas de planificación - Modelo de sistema de tiempo real: eventos periódicos - Análisis de planificabilidad - Protocolos de sincronización	10,00	0,00	10,00	0,00	0,00	2,50	2,50	5,00	20,00	0,00	0,00	6-10
TOTAL DE HORAS		30,00	0,00	30,00	0,00	0,00	7,50	7,50	15,00	60,00	0,00	0,00	

Esta organización tiene carácter orientativo.

TE	Horas de teoría
PA	Horas de prácticas en aula
PLE	Horas de prácticas de laboratorio experimental
PLO	Horas de prácticas de laboratorio en ordenador
CL	Horas de prácticas clínicas
TU	Horas de tutoría
EV	Horas de evaluación
TG	Horas de trabajo en grupo
TA	Horas de trabajo autónomo
TU-NP	Tutorías No Presenciales
EV-NP	Evaluación No Presencial

7. MÉTODOS DE LA EVALUACIÓN

Descripción	Tipología	Eval. Final	Recuper.	%
Programación paralela	Evaluación en laboratorio	No	Sí	33,33

Calif. mínima	4,00
Duración	
Fecha realización	Durante el curso
Condiciones recuperación	
Observaciones	Evaluación continua de las prácticas en el laboratorio. Se requerirá la presentación de una memoria sobre las prácticas desarrolladas. Recuperación en el periodo de recuperación establecido mediante examen en el laboratorio.

Programación concurrente y de tiempo real	Evaluación en laboratorio	No	Sí	66,67

Calif. mínima	4,00
Duración	
Fecha realización	Durante el curso
Condiciones recuperación	
Observaciones	Evaluación continua de las prácticas en el laboratorio: - Se evaluará el trabajo que realice individualmente el alumno en el laboratorio. Se requerirá la presentación de una memoria sobre las prácticas que haya desarrollado. La calificación de esta parte se obtendrá mediante la evaluación de al menos cuatro prácticas; en cada práctica evaluada se incluirá la evaluación del trabajo realizado en el laboratorio para la propia práctica y las anteriores que no hayan sido evaluadas. Recuperación en el periodo de recuperación establecido mediante examen en el laboratorio.

TOTAL	100,00

Observaciones

La calificación final se obtiene de acuerdo a las siguientes proporciones de los dos bloques temáticos:
- Programación paralela 1/3
- Programación concurrente y de tiempo real 2/3

Se podrá compensar uno de los bloques temáticos con una calificación mínima de 4.0. En caso de que uno de los bloques temáticos tenga una calificación inferior a 4.0, la nota final será el mínimo de 4.9 y la media obtenida. En caso de aprobar únicamente uno de los bloques temáticos en el periodo ordinario, se guardará la nota de esa parte para el periodo de recuperación.

Criterios de evaluación para estudiantes a tiempo parcial

Los alumnos a tiempo parcial que no puedan seguir las actividades de laboratorio y la evaluación continua propuesta, serán evaluados mediante pruebas equivalentes a las establecidas para el periodo de recuperación.

8. BIBLIOGRAFÍA Y MATERIALES DIDÁCTICOS

BÁSICA

PROGRAMACIÓN PARALELA

- Francisco Almeida, Domingo Giménez, José Miguel Mantas, Antonio M. Vidal: “Introducción a la Programación Paralela”. Editorial Paraninfo. 2008

PROGRAMACIÓN CONCURRENTE Y DE TIEMPO REAL

- Andy Wellings: "Concurrent and Real-Time Programming in Java". Wiley, 2004.
- J.S.W.Liu. “Real Time Systems”. Prentice Hall, 2000.
- JAVADOC, referencia de la Api J2SE 8.0. Disponible online.http://docs.oracle.com/javase/8/docs/api/

Complementaria

PROGRAMACIÓN PARALELA

- Barbara Chapman, Gabriele Jost, Ruud van van der Pas: "Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation)". The MIT Press, 2007.
- Maya Posch. Mastering C++ Multithreading. Packt Publishing, 2017.
- Miloš Ljumovic . C++ Multithreading Cookbook. Packt Publishing, 2014.

PROGRAMACIÓN CONCURRENTE Y DE TIEMPO REAL

- A. Burns and A. Wellings. “Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX”. Fourth Edition. Addison-Wesley, 2009.
- B. GOETZ: “Java concurrency in practice”. Addison Wesley, 2006.
- Scott Oaks, Henry Wong, Mike Loukides: “Java Threads (Java Series (O'Reilly & Associates))” O'Reilly & Associates books 3 edition, 2004.
- Doug Lea: “Concurrent Programming in Java(TM): Design Principles and Pattern” Prentice Hall PTR; 2 edition, 1999.
- Jeff Magee, Jeff Kramer: “Concurrency: State Models and Java Programs” John Wiley & Sons; 2 edition, 2006.
- M.H. Klein, T. Ralya, B. Pollak, R. Obenza, and M. González Harbour. “A practitioner’s Handbook for Real-Time Analysis”. Kluwer Academic Pub., 1993.
- G. Buttazzo. “Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications”. 3rd Ed., Springer, 2011.

9. SOFTWARE

PROGRAMA / APLICACIÓN	CENTRO	PLANTA	SALA	HORARIO
S.O. Linux (Java, C y MAST)	F. Ciencias		Lab. CTR	

10. COMPETENCIAS LINGÜÍSTICAS

- ☑ Comprensión escrita
- ☐ Comprensión oral
- ☐ Expresión escrita
- ☐ Expresión oral
- ☐ Asignatura íntegramente desarrollada en inglés

Observaciones

Parte del material bibliográfico recomendado o proporcionado está elaborado en inglés.