Universitat de Lleida

GUÍA DOCENTE

SISTEMAS CONCURRENTES Y PARALELOS

Coordinación: CORES PRADO, FERNANDO

Año académico 2016-17

SISTEMAS CONCURRENTES Y PARALELOS 2016-17

Información general de la asignatura

Denominación	SISTEMAS CONCURRENTES Y PARALELOS
Código	102022
Semestre de impartición	1R Q(SEMESTRE) EVALUACIÓN CONTINUADA
Carácter	

Grado/Máster	Curso	Carácter	Modalidad
Doble Titulación: Grado en Ingeniería Informática y Grado en Administración y Dirección de Empresas	4	OBLIGATORIA	Presencial
Grado en Ingeniería Informática	3	OBLIGATORIA	Presencial
Máster Universitario en Ingeniería Informática		COMPLEMENTOS DE FORMACIÓN	Presencial

Número de créditos ECTS	6
Grupos	1GG,2GM
Créditos teóricos	3
Créditos prácticos	3
Coordinación	CORES PRADO, FERNANDO
Departamento/s	INFORMATICA I ENGINYERIA INDUSTRIAL
Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante	6 ECTS = 25x6 = 150 horas de trabajo 40% --> 60 horas presenciales 60% --> 90 horas de trabajo autónomo del estudiante
Información importante sobre tratamiento de datos	Consulte este enlace para obtener más información.
Idioma/es de impartición	Preferentemente en Castellano, en Ingles si hay algún estudiante extranjero.
Distribución de créditos	Fernando Cores 9.0
Horario de tutoría/lugar	Lunes 16h-17h (s3/17) Jueves 12h-13h (s3/17)

SISTEMAS CONCURRENTES Y PARALELOS 2016-17

Profesor/a (es/as)	Dirección electrónica profesor/a (es/as)	Créditos impartidos por el profesor	Horario de tutoría/lugar
CORES PRADO, FERNANDO	fcores@diei.udl.cat	9	A concertar por correo electrónico

Información complementaria de la asignatura

La asignatura es inminentemente práctica, por lo que se dará más peso a las prácticas y la programación. Básicamente trabajaremos con dos lenguajes, C para la programación concurrente con hilos de ejecución y Java para introducir las Apis de alto nivel para la sincronización y los patrones concurrentes. Ninguno de estos lenguajes se explicará desde cero, debido a que tanto C como Java ya se han visto en asignaturas previas de la titulación. Para poder seguir la asignatura es imprescindible que los estudiantes tengan unos buenos fundamentos de programación en C y Java. En la asignatura se da por supuesto que los estudiantes son capaces de diseñar, desarrollar y depurar aplicaciones secuenciales de dificultad media sin muchos problemas.

Objetivos académicos de la asignatura

- Entender la importancia de la programación concurrente en las aplicaciones actuales.
- Identificar las principales características de los distintos tipos de sistemas concurrentes que existen.
- Conocer y entender los problemas que plantea el desarrollo de programas concurrentes y que no aparecen en la programación secuencial.
- Entender los conceptos de sincronización y exclusión mutua entre procesos.
- Identificar las propiedades de seguridad y vivacidad que un sistema concurrente debe cumplir y ser capaz de razonar si dichas propiedades se cumplen.
- Adquirir experiencia y conocimiento en los mecanismos de sincronización y comunicación que se utilizan en la actualidad para desarrollar programas concurrentes tanto para sistemas de memoria compartida como para sistemas distribuidos.
- Entender el funcionamiento de semáforos y monitores como mecanismos de sincronización para memoria compartida y comprender cómo se pueden resolver problemas de programación concurrente usando monitores.
- Aplicar metodologías de la Ingeniería del Software en el desarrollo de aplicaciones concurrentes y paralelas.

Competencias

Competencias transversales EPS:

- EPS7. Capacidad de trabajar en situaciones de falta de información y/o bajo presión.

Competencias transversales globales:

- GII-CRI11. Conocimiento y aplicación de las características, funcionalidades y estructura de los Sistemas Distribuidos, las Redes de Computadores e Internet y diseñar e implementar aplicaciones basadas en ellas.
- GII-CRI14. Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.
- GII-CRI16. Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

Contenidos fundamentales de la asignatura

1. **Introducción a la concurrencia**

1. Definición concurrencia
2. Necesidad y beneficios de la programación concurrente
3. Arquitecturas hardware concurrentes
4. Características sistemas concurrentes
5. Especificación programas Concurrentes
    1. Condiciones de Bernstein
    2. Tabla de concurrencia
    3. Grafos de precedencia
6. Formas de expresar la concurrencia
    1. Cobegin/Coend
    2. Fork/Join
    3. Unix+C
    4. Ejemplos
7. Caso de estudio: Hilos de Ejecución en Linux y Java

2. **Diseño aplicaciones concurrentes/paralelas**
    1. Modelo de programación concurrente/paralela
    2. Eficiencia Programas Concurrentes/Paralelos
        1. Alcance del paralelismo
        2. Granularidad
        3. Localidad
    3. Diseño programas concurrentes
        1. Etapas de Diseño
        2. Técnicas de Descomposición de tareas
        3. Patrones de diseño paralelos
        4. Distribución y comunicación de tareas
    4. Casos de estudio.
3. **Sincronización de procesos concurrentes**
    1. Introducción Sincronización
        1. Condiciones de Carrera
    2. El problema de la exclusión mutua
        1. Soluciones Software
        2. Soluciones Hardware
    3. Sincronización condicional
        1. Problemas clásicos de sincronización
    4. Caso de estudio:
        1. Pthread mutex y variables de condición
        2. Sincronización en Java.
4. **APIs de alto nivel para la concurrencia y sincronización**
    1. Monitores
    2. Patrones concurrencia
    3. Caso de estudio:
        1. Librería Boost
        2. Java Concurrent API

Ejes metodológicos de la asignatura

Grupos Grandes: Clases Teoría (3 créditos)

- Lección magistral: clases basadas en apuntes y transparencias en donde se presentarán los conceptos de la asignatura.
- Problemas: Los conceptos de la asignatura se trabajarán mediante una serie de problemas que se resolverán de forma colaborativa y que ayudarán a asimilar los conceptos clave.
- Casos de uso: Se aplicarán las técnicas vistas en clase a ejemplos reales y se analizará su impacto sobre el rendimiento de las aplicaciones.

Grupos Medios: Clases Laboratorio (3 créditos)

- Clases dirigidas y seguimiento personalizado por grupos de prácticas.
- Prácticas de laboratorio: se presentarán las tecnologías y APIs de programación concurrente y se trabajaran de forma práctica mediante tutoriales y ejemplos.
- Problemas: Realización y corrección de problemas relacionados tanto con la parte teórica como práctica de la asignatura.

Trabajo autónomo (no presencial):

- Las prácticas y los problemas se completarán fuera del horario de clase.

Plan de desarrollo de la asignatura

Sem	Descripción	Actividad Presencial GG	Actividad Presencial GM	Actividad trabajo autónomo
1	Presentación Introducción Concurrencia	Presentación Asignatura T1: Introducción Concurrencia	**L1: Posix Threads**	**Estudiar bibliografía y programa**
2	Introducción Concurrencia	T1: Introducción Concurrencia	**L1: Posix Threads** **L2: Java Threads** Práctica1: Presentación	T1: Introducción Concurrencia
3	Introducción Concurrencia	T1: Introducción Concurrencia T2: Diseño programas concurrentes	**L2: Java Threads** **FESTIVO GMA**	Práctica1 Problemas: Concurrencia
4	Diseño aplicaciones concurrentes	T2: Diseño aplicaciones concurrentes	Problema P1 Eval. Cont. Problemas: Introducción Concurrencia	Práctica 1 Problemas: Diseño
5	Diseño aplicaciones concurrentes	**FESTIVO**	Corrección P1 Eval. Cont.	Práctica 1 T2: Diseño
6	Diseño aplicaciones concurrentes	T2: Diseño aplicaciones concurrentes	Problemas: Diseño	Práctica 1 Problemas: Diseño
7	Diseño aplicaciones concurrentes Sincronización	T2: Diseño aplicaciones concurrentes T3: Sincronización procesos concurrentes	Práctica1: Entrega Problemas: Diseño	Práctica 1 T3: Sincronización
8	Sincronización	T3: Sincronización procesos concurrentes	Problemas: Sincronización	T3: Sincronización
9		**1er Parcial**		Estudiar
10	Sincronización	T3: Sincronización procesos concurrentes	**L3: Sincronización Linux** Práctica2: Presentación	Práctica 2 T3: Sincronización
11	Sincronización	T3: Sincronización procesos concurrentes	**L4: Sincronización Java**	Práctica 2 Problemas: Sincronización
12	Sincronización	T3: Sincronización procesos concurrentes	Problemas: Sincronización Problema P2 Eval. Cont.	Práctica 2 Problemas: Sincronización
13	APIs concurrencia y sincronización	T4: APIs de alto nivel para la concurrencia y sincronización	Práctica2: Entrega **FESTIVO GMA**	Problemas: Sincronización T4: APIs alto nivel

14	APIs concurrencia y sincronización	T4: APIs de alto nivel para la concurrencia y sincronización	Corrección P2 Eval. Cont. Práctica3: Presentación	Práctica 3 T4: APIs alto nivel
15	APIs concurrencia y sincronización	T4: APIs de alto nivel para la concurrencia y sincronización	Práctica3: Entrega	Práctica 3 T4: APIs alto nivel
16		**2^{on} Parcial**		Estudiar
17		**2^{on} Parcial**		Estudiar
18		TUTORIAS		
19		**Recuperación**		Estudiar

Sistema de evaluación

Tabla. Actividades de evaluación

Acr.	Actividad Evaluación	Ponderación	Nota Mínima	En grupo	Obligatoria	Recuperable
P1	*Examen 1er Parcial*	25%	4	NO	SI	SI
P2	*Examen 2^{on} Parcial*	25%	4	NO	SI	SI
PRA	*Prácticas*	40%	5	SI (<=2)	SI	1 Prác.
PRB	*Problemas*	10%	NO	SI (<=2)	NO	NO
PCL	*Participación Clase*	1 punto	NO	NO	NO	NO
Existe nota mínima de 4 en la media de las pruebas escritas para poder tener nota final de la asignatura. Se tienen que aprobar todas las prácticas de forma individual. Una práctica se considera suspendida si no llega a un 4.						
NotaFinal = 0,25*P1 + 0,25*P2 + 0,4*PRA + 0,1*PRB + 0,1*PCL						

La asignatura se aprueba con una nota final superior o igual 5 y habiendo realizado correctamente las prácticas de laboratorio (todas ellas con al menos un 4 de nota)

La nota final de la asignatura se obtiene a partir de la suma ponderada de las notas de los dos exámenes y las prácticas, más las notas de participación en clase y la evaluación continuada.

La asignatura tiene dos parciales, cada uno de ellos con un peso del 25% en la nota final. Estas pruebas son obligatorias y eliminan materia.

La realización y superación de las prácticas de laboratorio es obligatoria para aprobar la asignatura. Las prácticas se evaluarán con una nota que representará el 40% de la nota final de la asignatura. La copia de cualquiera de las prácticas de la asignatura implicará suspender todas las práctica y con ello la asignatura.

Existe nota mínima de 4 en la media de las pruebas escritas para poder obtener la nota final de la asignatura.

También existe nota mínima (5) para la nota media de las prácticas de laboratorio. Una práctica se considera suspendida si no llega a un 4.

Bibliografía y recursos de información

Bibliografía Básica:

- José Tomás Palma Méndez, M. C. Garrido Carrera, F. Sanchez Figueroa, A. Quesada Arencibia, "Programación Concurrente ", Thomson, 2003.
- Maurice Herlihy, Nir Shavit, "The Art of Multiprocessor Programming", Morgan Kaufmann, 2008
- Douglas Lea, "Concurrent Programming in Java: Design Principles and Patterns", Addison-Wesley Professional, 2000.

Bibliografía Ampliada:

- Gadi Taubenfeld, "Synchronization Algorithms and Concurrent Programming", Pearson / Prentice Hall, 2006
- M. Ben-Ari, "Principles of Concurrent and Distributed Programming", Addison-Wesley, 2nd Edition, 2006
- KayA. Robbins, Steven Robbins, "UNIX Programación Práctica. Guía para la Concurrencia, la Comunicación y los Multihilos", Edt.Prentice-Hall, 1997.