

CENTRO UNIVERSITARIO DE TECNOLOGÍA Y ARTE DIGITAL



PLANIFICACIÓN DE LA DOCENCIA UNIVERSITARIA

GUÍA DOCENTE

Introducción a la concurrencia

1. DATOS DE IDENTIFICACIÓN DE LA ASIGNATURA.

Título:	Grado en Ingeniería de Desarrollo de Contenidos Digitales				
Facultad:					
Departamento/Instituto:					
Materia:	Fundamentos de Sistemas Software y Hardware				
Denominación de la asignatura:	Introducción a la concurrencia				
Código:	0048025				
Curso:	3				
Semestre:	2				
Tipo de asignatura (básica, obligatoria u optativa):	obligatoria				
Créditos ECTS:	6				
Modalidad/es de enseñanza:	Presencial				
Lengua vehicular:	Castellano				
Equipo docente:	Marcos Novalbos Mendiguchía				
Profesor/a:	Marcos Novalbos Mendiguchía				
Grupos:	IDCD3				
Despacho:	Sala de profesores				
Teléfono:	916402811	Ext.	113	E-mail:	Marcos.novalbos@live.u-tad.com
Página web:	u-tad.blackboard.com				

2. REQUISITOS PREVIOS.

Esenciales:
Haber cursado Programación a Bajo Nivel
Aconsejables:
Haber aprobado Introducción a la Programación, Estructuras de los Ordenadores y Programación a Bajo Nivel

3. SENTIDO Y APORTACIONES DE LA ASIGNATURA AL PLAN DE ESTUDIOS.

Campo de conocimiento al que pertenece la asignatura.
Esta asignatura pertenece al Módulo de Fundamentos, a la Materia de Fundamentos de sistemas software y hardware.
Relación de interdisciplinariedad con otras asignaturas del currículum.
Esta asignatura partiendo de la base de Introducción a la Programación, Algoritmos y Estructuras de Datos, continúa con la enseñanza de programación a un nivel cercano al hardware, especializado en la programación de sistemas concurrentes y en la programación en paralelo, conceptos que entroncan con la optativa de cuarto curso de Sistemas Distribuidos.
Aportaciones al plan de estudios e interés profesional de la asignatura.
Esta asignatura es relevante para adquirir los conceptos de programación paralela, concurrencia entre varios threads, sincronización y computación distribuída.

4. RESULTADOS DE APRENDIZAJE EN RELACIÓN CON LAS COMPETENCIAS QUE DESARROLLA LA ASIGNATURA.

COMPETENCIAS ESPECÍFICAS	RESULTADOS DE APRENDIZAJE RELACIONADOS CON LAS COMPETENCIAS ESPECÍFICAS
<p>CE12 - manifestar 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</p> <p>CE14 - Tener conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente y distribuida</p>	<p>Resolver problemas relacionados con la gestión de recursos (procesos, memoria, dispositivos de E/S y ficheros) de un sistema operativo.</p> <p>Comprender las dificultades y las estrategias de la programación de sistemas concurrentes.</p>

5. CONTENIDOS /TEMARIO / UNIDADES DIDÁCTICAS

Tema 1: Introducción al rendimiento de los computadores

En este tema se repasarán los fundamentos básicos para poder medir el rendimiento de dos computadores distintos, analizar las prestaciones de cada uno de ellos y obtener un resultado comparativo de rendimiento. Se presentará la “ley de Amdahl” y los conceptos de Speedup como sistema de comparación de rendimiento.

Tema 2: Procesadores segmentados y superescalares

Se presentará el concepto de segmentación como técnica de optimización de algoritmos. Se usará como caso de estudio los procesadores segmentados MIPS, para demostrar cómo se puede optimizar y paralelizar la ejecución a nivel de instrucción en sistemas uniprosesor.

Tema 3: Multiprocesadores/ GPUs

Se presenta el concepto de hilo de ejecución para paralelización a alto nivel. Se hará un estudio de diversas arquitecturas CPU y GPU multiprosesor para evaluar las técnicas disponibles.

Tema 4: Clusters de computadores personales/GPUs

Se presentará el concepto de cluster de ordenadores, y las ventajas de usar ese sistema de computación.

Tema 5: Programación de Multiprocesadores

Tema orientado a las librerías de programación para sistemas multiprosesor/gpus

Tema 6: Programación paralela basada en paso de mensajes (Clusters)

Tema orientado a las librerías de programación para sistemas cluster/multigpu

Práctica Multiprocesadores/threads

Primer ejercicio práctico que durará varias sesiones, realizado en el laboratorio/aula de ordenadores. Se presentará un problema que deberá de ser resuelto usando los conocimientos obtenidos en teoría. Se realizará en paralelo con el resto de temas de teoría

Práctica Clusters/GPUs

Segundo ejercicio práctico que durará varias sesiones, realizado en el laboratorio/aula de ordenadores. Se presentará un problema que deberá de ser resuelto usando los conocimientos obtenidos en teoría. Se realizará en paralelo con el resto de temas de teoría

6. CRONOGRAMA

UNIDADES DIDÁCTICAS / TEMAS	PERÍODO TEMPORAL
Tema 1: Introducción al rendimiento de los computadores	9/2/2016-15/2/2016
Tema 2: Procesadores segmentados y superescalares	15/2/2016-16/2/2016
Tema 3: Multiprocesadores/ GPUs	22/2/2016-7/3/2016
Práctica Multiprocesadores/threads	8/3/2016-4/4/2016
Tema4: Clusters de computadores personales/GPUs	4/4/2016-11/4/2016
Tema 5: Programación de Multiprocesadores	12/4/2016-18/4/2016
Tema 6: Programación paralela basada en paso de mensajes (Clusters)	19/4/2016-26/4/2016
Práctica Clusters/GPUs	03/5/2016-17/5/2016

7. MODALIDADES ORGANIZATIVAS Y MÉTODOS DE ENSEÑANZA

MODALIDAD ORGANIZATIVA	MÉTODO DE ENSEÑANZA	COMPETENCIAS RELACIONADAS	HORAS PRESENCIALES	TRABAJO AUTÓNOMO	TOTAL DE HORAS
Clases teóricas	Lección magistral	CE12, CE14	21	1	22
Seminarios y talleres	Estudio de casos Resolución de ejercicios y problemas		0	0	0
Clases prácticas	Aprendizaje basado en problemas Aprendizaje orientado a proyectos	CE12, CE14	22	0	22
Prácticas externas			0	0	0
Tutorías	Aprendizaje orientado a proyectos Aprendizaje basado en problemas	CE12, CE14	8	0	8
Actividades de evaluación		CE12, CE14	8	0	8
Estudio y trabajo en grupo	Aprendizaje cooperativo	CE12, CE14	1	21	23
Estudio y trabajo autónomo, individual	Estudio de casos Resolución de ejercicios y problemas Aprendizaje basado en problemas Aprendizaje orientado a proyectos	CE12, CE14	0	68	68

8. SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CRITERIOS DE EVALUACIÓN	VALORACIÓN RESPECTO A LA CALIFICACIÓN FINAL (%)
Examen Parcial 1	Examen teórico, evaluación del conocimiento adquirido en las clases de teoría para los temas los temas 1-3. Obligatorio sacar al menos una nota de 5 sobre 10 para aprobar.	15%
Examen Parcial 2	Examen teórico, evaluación del conocimiento adquirido en las clases de teoría para los temas los temas 4-6. Obligatorio sacar al menos una nota de 5 sobre 10 para aprobar.	15%
Examen Práctico 1: Memoria compartida	Evaluación de la práctica propuesta durante las sesiones de laboratorio para la sección de Memoria compartida (threads). Obligatorio sacar al menos una nota de 5 sobre 10 para aprobar.	30%
Examen Práctico 2: Memoria distribuida	Evaluación de la práctica propuesta durante las sesiones de laboratorio para la sección de Memoria distribuida (cluster/MPI). Obligatorio sacar al menos una nota de 5 sobre 10 para aprobar.	30%
Ejercicios complementarios	Evaluación de los ejercicios complementarios pedidos durante las sesiones teóricas. Opcional para aprobar.	10%

9. BIBLIOGRAFÍA / WEBGRAFÍA

Bibliografía básica (1-3 libros)

Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation)

Parallel Programming in C with MPI and OpenMP

CUDA C BEST PRACTICES GUIDE (Cuda 6.5)

Bibliografía recomendada (Max 10 libros)

10.- MATERIAL, SOFTWARE Y HERRAMIENTAS NECESARIAS

TIPOLOGÍA DEL AULA:

Aula de ordenadores, ordenadores multicore conectados en red con posibilidad de comunicación a través de MPI/sockets. Capacidad para arrancar con imágenes de sistema operativo linux desde Pendrive/usb

MATERIALES:

Al menos un ordenador por alumno conectado en red con posibilidad de comunicación a través de MPI/sockets. Cada ordenador:

- Posibilidad de elegir unidad desde la que arrancar (USB ó disco duro interno).
- Una GPU NVidia Serie GTX/quadro/etc... o superior que soporte al menos CUDA 4.0.
- Al menos dos procesadores (dual core) o superior
- IP estática o de subred accesible desde cada ordenador del aula

SOFTWARE:

- Sistema operativo Ubuntu/Linux 12.04 x86-64 o superior (14.04)
- CudaToolkit SDK para Linux (driver de cuda para nvidia, nsight y ejemplos del toolkit) :

<https://developer.nvidia.com/cuda-downloads>

- Paquete de compilación paraggcc/g++/make/cmake.
- Librerías OpenMPI para linux (instalables desde repositorio):
- Librerías gráficas freeglut de desarrollo (libfreeglut3-dev)
- Eclipse IDE para C/C++
- Máquina virtual de Java con soporte RMI