



Teaching Guide

Identifying Data					2015/16
Subject (*)	Arquitectura e Enxeñaría de Computadores		Code	614111401	
Study programme	Enxeñeiro en Informática				
Descriptors					
Cycle	Period	Year	Type	Credits	
First and Second Cycle	Yearly	Fourth	Troncal	9	
Language	SpanishGalician				
Teaching method	Face-to-face				
Prerequisites					
Department	Electrónica e Sistemas				
Coordinador	Doallo Biempica, Ramon	E-mail	ramon.doallo@udc.es		
Lecturers	Doallo Biempica, Ramon	E-mail	ramon.doallo@udc.es		
Web	campusvirtual.udc.es/moodle/				
General description	- Arquitecturas microprocesador (microprocesadores comerciais, multithreading, multicore, streaming) - Arquitecturas multiprocesador				

Study programme competences / results

Code	Study programme competences / results
A1	Aprender de maneira autónoma novos coñecementos e técnicas avanzadas axeitadas para a investigación, o deseño e o desenvolvemento de sistemas e servizos informáticos.
A2	Concibir e desenvolver novas arquitecturas de computación, en especial para sistemas multiprocesadores, analizando e adaptando diversas alternativas tecnolóxicas a cada problema concreto.
A3	Concibir e planificar o desenvolvemento de aplicacións informáticas complexas ou con requisitos especiais.
A4	Coñecer e aplicar diferentes protocolos de comunicación e sistemas de xestión de rede.
A6	Avaliar, definir, seleccionar e auditar plataformas hardware e software para a execución e desenvolvemento de aplicacións e servizos informáticos.
A8	Concibir, despregar, organizar e xestionar un servizo informático complexo.
B2	Resolver problemas de forma efectiva.
B3	Aplicar un pensamento crítico, lóxico e creativo.
B5	Traballar de forma colaborativa.
B8	Traballar en equipos de carácter interdisciplinar.
B9	Capacidade para tomar decisións.
B11	Razoamento crítico.
B12	Capacidade para a análise e a síntese.
B13	Capacidade de comunicación.
B15	Motivación pola calidade.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.

Learning outcomes

Learning outcomes	Study programme competences / results		
Utilizar novas arquitecturas de computación, tanto microprocesador como multiprocesador, en solucións en enxeñaría informática.	A1 A2 A3 A6 A8	B2	C6



Configurar sistemas para servidores y sistemas de alto rendimiento.	A2 A3 A8	B3	
Analizar de forma crítica parámetros de rendimiento de sistemas hardware y aplicaciones específicas con necesidades especiales.	A2 A3 A4 A6 A8	B9 B11 B12	C6
Desarrollar aplicaciones paralelas en equipo siguiendo el método de ingeniería informática.	A2 A6	B2 B5 B8 B11 B12 B13 B15	C6

Contents	
Topic	Sub-topic
1. Introducción al procesamiento paralelo	1.1. Procesamiento paralelo 1.2. Condiciones de paralelismo - Concepto de dependencia - Condiciones de Bernstein - Paralelismo hardware y software 1.3. Niveles de paralelismo 1.4. Importancia del procesamiento paralelo 1.5. Clasificaciones de arquitecturas paralelas - Taxonomía de Flynn - Organización del sistema memoria: clasificación tradicional de sistemas MIMD. 1.6. Medidas de rendimiento



<p>2. Paralelismo a nivel de instrucción</p>	<p>2.1. Rendimiento de un procesador</p> <p>2.2. Soluciones básicas para la mejora del rendimiento: Evolución</p> <p>2.3. Técnicas para el aumento del paralelismo a nivel de instrucción</p> <p>2.4. Planificación estática/software</p> <ul style="list-style-type: none">- Planificación estática básica- Desenrollamiento de bucles- Planificación estática superescalar- Planificación estática VLIW- Segmentación software- Detección y eliminación de dependencias <p>2.5. Planificación dinámica/hardware</p> <ul style="list-style-type: none">- Planificación dinámica superescalar <p>2.6. Renombre de registros</p> <ul style="list-style-type: none">- Renombre de registros software- Renombre de registros hardware: Buffer de reordenamiento <p>2.7. Tratamiento de operaciones load/store: buffer de almacenamiento</p> <p>2.8. Tratamiento de riesgos de control</p> <ul style="list-style-type: none">- Predicción de salto estática, salto retardado- Predicción dinámica de saltos <p>2.9. Especulación</p> <p>2.10. Ejemplos comerciales</p>
<p>3. Procesamiento vectorial</p>	<p>3.1. ¿Por qué máquinas vectoriales?</p> <p>3.2. Arquitecturas vectoriales</p> <p>3.3. Arquitectura vectorial básica</p> <ul style="list-style-type: none">- Componentes principales de la U.V.- Máquinas vectoriales comerciales- Repertorio de instrucciones de la U.V.- Ejemplo de bucle vectorial <p>3.4. Vectorización</p> <p>3.5. Tiempo de ejecución vectorial</p> <p>3.6. Organización de memoria en un procesador vectorial</p> <ul style="list-style-type: none">- Entrelazamiento de memoria- Técnicas de estructuración de datos para máquinas vectoriales <p>3.7. Control de la longitud del vector</p> <p>3.8. Rendimiento vectorial</p> <ul style="list-style-type: none">- Efectos de la vectorización.- Mejorando el rendimiento vectorial:<ul style="list-style-type: none">? Encadenamiento de U.F. vectoriales? Transformación de código para evitar conflictos de acceso a memoria <p>3.9. Ejemplos comerciales:</p> <ul style="list-style-type: none">- Earth Simulator- Cray X1



4. Computación paralela: arquitecturas y paradigmas de programación	<p>4.1. Introducción</p> <p>4.2. Arquitecturas paralelas modernas:</p> <ul style="list-style-type: none"> * Arquitecturas paralelas vectorial, multimedia y GPU * Arquitecturas paralelas TLP: multiprocesador y multicomputador * Arquitecturas paralelas RLP: cluster y WSC * Arquitecturas paralelas híbridas * Comparación entre arquitecturas paralelas modernas <p>4.4. Paradigmas de programación</p> <ul style="list-style-type: none"> * Paradigma de memoria compartida, memoria distribuida, PGAS, streaming y MapReduce. * Paradigmas híbridos * Comparación entre herramientas disponibles en el mercado * Comparación entre soluciones tecnológicas disponibles en el mercado
5. Arquitecturas multiprocesador: coherencia caché y consistencia de memoria	<p>5.1. Introducción</p> <p>5.1. Sincronización</p> <p>5.2. Modelos de consistencia de memoria</p> <p>5.2. Protocolos de coherencia caché</p> <ul style="list-style-type: none"> * Protocolo de snooping * Protocolo basado en directorio
6. Redes de interconexión para arquitecturas paralelas	<p>6.1. Introducción</p> <p>6.2. Redes de interconexión estáticas</p> <p>6.3. Estrategias de conmutación</p> <p>6.4. Redes de interconexión dinámicas</p> <p>6.5. Funciones de encaminamiento de datos</p> <p>6.6. On-chip networks (OCN)</p>

Planning				
Methodologies / tests	Competencies / Results	Teaching hours (in-person & virtual)	Student?s personal work hours	Total hours
Objective test	A1 A2 A3 A4 A6 A8 B2 B3 B5 B8 B9 B11 B12 B13 B15 C6	4	0	4
Personalized attention		3	0	3

(*)The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.

Methodologies	
Methodologies	Description
Objective test	Prueba sobre los conceptos teóricos presentados en las sesiones magistrales.

Personalized attention	
Methodologies	Description
Objective test	<p>Prueba objetiva:</p> <ul style="list-style-type: none"> - Atención personal en horas de tutoría para aclarar dudas sobre los contenidos teórico/prácticos de la materia. - Revisión de los ejercicios realizados.



Assessment

Methodologies	Competencies / Results	Description	Qualification
Objective test	A1 A2 A3 A4 A6 A8 B2 B3 B5 B8 B9 B11 B12 B13 B15 C6	Prueba sobre los contenidos trabajados en las sesiones magistrales.	100
Others			

Assessment comments

- La evaluación de los alumnos consistirá en una prueba escrita en las fechas indicadas en el calendario académico oficial. - En los exámenes de todas las convocatorias se aplicará el mismo criterio en cuanto a que el alumno debe obtener al menos la mitad de la nota para superar la asignatura.

Sources of information

Basic	<ul style="list-style-type: none"> - Ortega, J., Anguita, M. y Prieto, A. (2005). Arquitectura de Computadores. Thomson - Hennessy, J. L. y Patterson, D. A. (2012). Computer Architecture: A Quantitative Approach (5th edition). Morgan Kaufmann - Patterson, D. A. y Hennessy, J. L. (2000). Estructura y Diseño de Computadores. Reverté - Stallings, W. (2006). Organización y arquitectura de computadores. Reverté
Complementary	<ul style="list-style-type: none"> - Sima, D. , Fountain,T. y Kacsuk, P. (1997). Advanced Computer Architecture. Addison-Wesley - Culler, D. E. y Singh,J. P. (1999). Parallel Computer Architecture: a Hardware/Software Approach. Morgan Kaufmann - Hwang, K. y Xu, Z. (1998). Scalable Parallel Computing. McGraw-Hill - Tannebaum, A. S. (1999). Structured Computer Organization. Prentice Hall

Recommendations

Subjects that it is recommended to have taken before

Estrutura de Computadores I/614111208
Sistemas Operativos II/614111302
Estrutura de Computadores II/614111306
Redes de Comunicaci3n/614111307

Subjects that are recommended to be taken simultaneously

Arquitecturas Tolerantes a Fallos/614111605
Optimizaci3n do Procesamento Paralelo/614111630

Subjects that continue the syllabus

Arquitecturas Tolerantes a Fallos/614111605
Optimizaci3n do Procesamento Paralelo/614111630

Other comments

(*)The teaching guide is the document in which the URV publishes the information about all its courses. It is a public document and cannot be modified. Only in exceptional cases can it be revised by the competent agent or duly revised so that it is in line with current legislation.