



Universitat de Lleida

GUÍA DOCENTE

SISTEMAS CONCURRENTES Y PARALELOS

Año académico 2013-14

Información general de la asignatura

Denominación	SISTEMAS CONCURRENTE Y PARALELOS
Código	102022
Semestre de impartición	1r Q Avaluació Continuada
Carácter	Obligatòria
Número de créditos ECTS	6
Créditos teóricos	3
Créditos prácticos	3
Departamento/s	Informàtica i Enginyeria Industrial
Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante	60 horas presenciales 90 horas trabajo autónomo
Información importante sobre tratamiento de datos	Consulte este enlace para obtener más información.
Idioma/es de impartición	Castellano
Distribución de créditos	Fernando Cores 6.0
Horario de tutoría/lugar	Lunes 16h-17h (s3/17) Jueves 12h-13h (s3/17)

Información complementaria de la asignatura

Asignatura/materia en el conjunto del plan de estudios (máx. 4000 caracteres)

La industria informática está sufriendo, si no otra revolución, sin duda una vigorosa agitación. Los principales fabricantes de procesadores han renunciado, por el momento, a tratar de diseñar/fabricar procesadores que funcionen más rápido. La Ley de Moore todavía no ha sido derogada: cada año, más y más transistores caben en un mismo espacio, pero su velocidad de reloj no se puede aumentar debido al sobrecalentamiento y al alto consumo asociado con las altas frecuencias de reloj. En lugar de incrementar la velocidad de los procesadores, los fabricantes han optado por las arquitecturas "multi-núcleo", en las que múltiples procesadores (núcleos) se integran conjuntamente dentro de un chip, los cuales se pueden comunicar directamente a través de memorias cache compartidas. Los chips multiprocesador realizan los cálculos de forma más eficiente mediante la explotación de paralelismo: el aprovechamiento de múltiples procesadores para trabajar en una única tarea. La generalización de las arquitecturas multiprocesador tendrá un impacto generalizado sobre cómo se diseña y desarrolla de software futuro. Hasta hace poco, los avances en la tecnología implicaba también avances en la velocidad de reloj, por lo que el software aceleraba efectivamente su velocidad de ejecución a media que pasaba el tiempo. Ahora, Sin embargo, esta mejora automática de las aplicaciones ha terminado. Los avances en la tecnología significarán un mayor paralelismo y no aumentar la velocidad de reloj, por lo tanto, la explotación de este paralelismo es una de los mayores retos pendientes de la informática moderna. Esta asignatura se centra en el diseño de aplicaciones concurrentes y paralelas capaces aprovechar de toda la capacidad de procesamiento de los sistemas multiprocesador para mejorar el rendimiento de las aplicaciones. Al desarrollar estas aplicaciones surgen desafíos de programación a todos los niveles del sistema multiprocesador - a una escala muy pequeña, dentro de los núcleos de un mismo chip, se necesita coordinar el acceso a una misma posición de memoria compartida, y a mayor escala, se necesita coordinar el enrutamiento de los datos entre los procesadores en superordenador. La programación concurrente y paralela representa un desafío mayor debido a que los modernos sistemas de computación son inherentemente asíncronos: las actividades se puede detener o retrasar sin previo aviso por las interrupciones, preferencia, fallos de caché, las excepciones, y otros eventos. Estos retrasos son inherentemente impredecibles, y puede variar enormemente en duración, provocando que el comportamiento de una aplicación concurrente/paralela pueda no ser determinista (diferentes ejecuciones con los mismos parámetros de entrada pueden generar distintos resultados) si no se programa correctamente. En este curso, vamos a estudiar las técnicas de programación concurrente, especialmente adecuadas para obtener el máximo rendimiento de los sistemas multiprocesador. También vamos a examinar algunos lenguajes paralelos y bibliotecas que permiten a los programadores aprovechar las prestaciones de múltiples procesadores u ordenadores. Vamos a centrarnos en los diferentes modelos de expresar la concurrencia, la utilización de los hilos de ejecución para el diseño de aplicaciones concurrentes, los mecanismos de sincronización tanto de bajo nivel (semáforos) como de alto nivel (monitores y variables de condición) para coordinar las tareas de una aplicación concurrente, lenguajes de programación paralela y diferentes APIs de alto nivel que faciliten la programación de estas aplicaciones.

Objetivos académicos de la asignatura

Ver competencias

Competencias

Competencias específicas de la titulación

- Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

Objetivos

- Aplicar metodologías de la Ingeniería del Software en el desarrollo de aplicaciones concurrentes y paralelas.
- Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

Objetivos

- Sin Traducir - Conocer los principales modelos de programación concurrente, paralela y distribuida.
- Sin Traducir - 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.
- Sin Traducir - 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.
- Sin Traducir - Entender la importancia de la programación concurrente en las aplicaciones actuales.
- Sin Traducir - Identificar las principales características de los distintos tipos de sistemas concurrentes que existen.
- Sin Traducir - Conocer y entender los problemas que plantea el desarrollo de programas concurrentes y que no aparecen en la programación secuencial.
- Sin Traducir - Entender los conceptos de sincronización y exclusión mutua entre procesos.

Competencias transversales de la titulación

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

Objetivos

- Sin Traducir - Identificar las propiedades de seguridad y vivacidad que un sistema concurrente debe cumplir y ser capaz de razonar si dichas propiedades se cumplen.

Contenidos fundamentales de la asignatura

Bloc I. Sistemas Concurrentes.

1. Introducción a la concurrencia

- 1.1. Procesos cooperativos
- 1.2. Concurrencia
- 1.3. Formas de expresar la concurrencia

- 1.4. Casode estudio: Hilos de Ejecución en Linux y Java

2. Sincronización de procesos concurrentes

- 2.1. Sincronización
- 2.2. El problema de la exclusión mútua
 - 2.2.1 Soluciones Software
 - 2.2.2 Soluciones Hardware
- 2.3. Semáforos
- 2.4. Problemas clásicos de sincronización
- 2.5. Caso de estudio:
 - 2.5.1 Semáforos IPC (Inter-Process Communication)
 - 2.5.2 Pthread mutex y variables de condición

3. APIs de alto nivel para la concurrencia y sincronización

- 3.1. Patrones concurrentes
- 3.2. Monitores
- 3.3. Caso de estudio:
 - 3.3.1 Concurrencia en Java
 - 3.3.2 Sincronización en Java

Bloc II. Sistemas paralelos.

4. Introducción a la computación paralela

- 4.1. ¿Porqué es necesaria la computación paralela
- 4.2. Hardware paralelo
- 4.3. Software paralelo
- 4.4. Rendimiento
 - 4.4.1 Speedup y eficiencia
 - 4.4.2 La ley de Amdahl
 - 4.4.3 Sobrecoste del paralelismo.

5. Programación paralela con MPI

- 5.1. Conceptos Básicos

5.2. Diseño programas paralelos

5.2.1 Paradigmas programas paralelos

5.3. Introducción a MPI (Message Passing Interface)

5.3.1 Instalación y configuración

5.3.2 Estructura básica

5.3.3 Comunicaciones

5.3.4 Operaciones colectivas

5.3.5 Ejemplos

Plan de desarrollo de la asignatura

Sem	Descripció:	Activitat Presencial GG	Activitat Presencial (GM)	HTP (6 Hrs)	Activitat treball autònom	HTNP (9 Hrs)
1	Introducció Concurrència	Presentació Assignatura T1: Introducció Concurrència	L1: Posix Threads	4	T1: Introducció Concurrència	6
2	Introducció Concurrència	T1: Introducció Concurrència	L2: Java Threads Pràctica 1: Presentació	4	T1: Introducció Concurrència	6
3	Disseny aplicacions concurrents	T2: Disseny aplicacions concurrents	L2: Java Threads Problemes: Disseny	4	Pràctica 1 T2: Disseny	5
4	Disseny aplicacions concurrents	T2: Disseny aplicacions concurrents	Problemes: Disseny	4	Pràctica 1 Problemes: Disseny	6
5	Disseny aplicacions concurrents	T2: Disseny aplicacions concurrents	Problemes: Disseny	4	Pràctica 1 Problemes: Disseny	5
6	Sincronització	T3: Sincronització processos concurrents	Pràctica 1: Entrega Correcció P1 Eval. Cont.	4	Problemes: Disseny	6
7	Sincronització	T3: Sincronització processos concurrents Repàs 1er Parcial	L3: Sincronización Linux Pràctica 2: Presentació	4	T3: Sincronització	7
8		1er Parcial			Estudiar	6
9	Sincronització	Correcció Parcial 1 T3: Sincronització processos concurrents	L4: Sincronización Java	4	Pràctica 2 T3: Sincronització	4

10	Sincronització	T3: Sincronització processos concurrents	Problemes: Sincronització	4	Práctica2 Problemes: Sincronització	8
11	Sincronització	T3: Sincronització processos concurrents	Práctica2: Entrega	4	Práctica2 Problemes: Sincronització	7
12	APIs concurrència i sincronització	T4:API d'alt nivell per a la concurrència i sincronització	Práctica3: Presentació Problemes: Sincronització	4	Problemes: Sincronització T4: APIs alt nivell	7
13	APIs concurrència i sincronització	T4:API d'alt nivell per a la concurrència i sincronització	Correcció P2 Eval. Cont.	4	T4: APIs alt nivell Práctica3	7
14	Optimització programes concurrents	T5: Optimització programes concurrents	Práctica3: Entrega Práctica4: Presentació	4	Práctica3 Práctica4	7
15	Optimització programes concurrents	T5: Optimització programes concurrents Repàs 2nd Parcial	Práctica4: Entrega	4	Práctica4	7
16		2^{on} Parcial		2	Estudiar	6
17						
18						
19		Recuperació		2		5

Sistema de evaluación

Actividad de Evaluación	Ponderación	Nota Mínima	En grupo	Obligatoria
<i>Examen 1^{er} Parcial</i>	25%	NO	NO	SI
<i>Examen 2^{on} Parcial</i>	25%	NO	NO	SI
<i>Prácticas</i>	40%	5	SI (≤ 2)	SI
<i>Problemas</i>	10%	NO	SI (≤ 2)	NO
<i>Participación Clase</i>	1 punto	NO	NO	NO

Bibliografía y recursos de información

Bibliografía Básica:

- [Pal03] José Tomás Palma Méndez, M. C. Garrido Carrera, F. Sanchez Figueroa, A. Quesada Arencibia, "Programación Concurrente ", Thomson, 2003.
- [Her08] Maurice Herlihy, Nir Shavit, "The Art of Multiprocessor Programming", Morgan Kaufmann, 2008.
- [Dou00] Douglas Lea, "Concurrent Programming in Java: Design Principles and Patterns", Addison-Wesley Professional, 2000.
- [Pac11] Peter Pacheco, "An Introduction to Parallel Programming", Morgan Kaufmann, 2011.

Bibliografía Ampliada:

- [Gad06] Gadi Taubenfeld, "Synchronization Algorithms and Concurrent Programming", Pearson / Prentice Hall, 2006
- [Ben06] M. Ben-Ari, "Principles of Concurrent and Distributed Programming", Addison-Wesley, 2nd Edition, 2006
- [Nic96] B. Nichols, D. Buttlar, "Pthreads Programming", O'Reilly, 1996
- [Kay97] Kay A. Robbins, Steven Robbins, "UNIX Programación Práctica. Guía para la Concurrencia, la Comunicación y los Multihilos", Edt. Prentice-Hall, 1997.
- [Afz97] Afzal, Amir, "Introducción a UNIX un enfoque práctico ", Edt. Prentice-Hall,