

PROGRAMACIÓN CONCURRENTE Y DISTRIBUIDA

3º Curso de Ingeniería Informática

Curso 2011/2012

[http://www.ctr.unican.es/asignaturas/procodis_3_II/index.html]

Profesores: José María Drake drakej@unican.es

Laura Barros barrosl@unican.es

Miguel Tellería de Esteban telleriam@unican.es

Objetivos de la asignatura:

Es formar al alumno para que sea capaz de concebir, especificar, diseñar, implementar y verificar aplicaciones informáticas en la que se utilice programación concurrente y programación distribuida. Se proporcionan 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. Se forma al alumno para que pueda implementar programas con concurrencia virtual sobre un monoprocesador, como los basados en un lenguaje de programación concurrente como Java, o bien programación con concurrencia física como la basada en plataformas distribuidas.

Son objetivos concretos de esta asignatura:

- Conocer y practicar la programación concurrente, así como identificar sus ventajas y problemas en contraste con la programación secuencial.
- Conocer los métodos formales para especificar, analizar y verificar programas concurrentes.
- Conocer y practicar la programación concurrente en Java.
- Conocer los principales recursos de comunicación, de sistemas operativos y de middleware de distribución en que se basa los sistemas distribuidos basados en plataformas heterogéneas.
- Conocer y practicar el diseño y la implementación de aplicaciones distribuidas basadas en RMI Java y en el middleware ICE de tipo CORBA.

Aptitudes destrezas

- El alumno debe ser capaz de concebir una aplicación como concurrente, a fin de decidir si presenta ventajas frente a la concepción secuencial de la misma.
- Debe conocer las alternativas de implementar una aplicación concurrente tanto en un entorno monoprocesador utilizando un lenguaje de programación, o los recursos de los sistemas operativos, o una plataforma distribuida.
- Debe saber programar aplicaciones concurrentes utilizando threads Java.
- Debe ser capaz de concebir un sistema como distribuido, a fin de decidir si presenta ventajas frente a una concepción centralizada.
- Debe saber implementar una aplicación distribuida utilizando RMI Java y ICE.
- Saber integrar la introducción de los niveles de concurrencia en el proceso de desarrollo de una aplicación informática, como medio de simplificar la estructura interna, optimizar el uso de los recursos disponibles, mejorar su respuesta temporal y hacerlas mas robustas frente a fallos.

PROGRAMA DE LA ASIGNATURA

a) Teoría: 4.5 Créditos (45 horas)

- 1 **Introducción a la programación concurrente.** (3 h)
Necesidad y ámbito de aplicación. Procesos. Mecanismos de soporte de la concurrencia
Patologías específicas de programación concurrente.
2. **Sincronización en programación concurrente.** (4 h.)
Modelos basados en intercambio de mensajes: Modelos basados en memoria compartida: Semáforos, regiones críticas y monitores. Completitud y expresividad de los mecanismos sincronización.
3. **Modelo Java de programación concurrente.** (6 h)
Creación, propiedades y ciclo de vida de los hilos de concurrencia. Mecanismos de sincronización entre hilos concurrentes. Jerarquía de hilos. Patrones de concurrencia implementados en Java.
4. **Arquitecturas y patrones para aplicaciones concurrentes** (4 h)
Patrones de concurrencia. Arquitecturas para aplicaciones concurrentes.
5. **Métodos formales de expresión de la concurrencia.** (3h)
Álgebra de procesos. Redes de Petri. Análisis de patologías.
- 6 **Programación distribuida.** (4 h)
Necesidad y ámbitos de aplicación. Modelos de programación distribuida. Soporte de la programación distribuida. Redes de comunicación, sistemas operativos y middleware de distribución. Transacciones. Servicios de soporte de la programación distribuida.
- 7 **Java distribuido RMI** (8 h)
Modelo Java de programación distribuida. Clases e interfaces Java para RMI. Invocación de servicios. Eventos y notificaciones. Soporte de red.
- 8 **Programación distribuida con objetos y componentes.** (8 h)
Arquitecturas basadas en objetos distribuidos. Lenguajes de descripción de las interfaces. Correspondencia con los lenguajes de programación. Servicios distribuidos. Referencia a objetos. Clases e interfaces Java para programación basada en objetos distribuidos.

b) Proyectos en laboratorio: 3 créditos (30 horas).

Seminario I: Descripción de la concurrencia y distribución con UML (3 horas)

Aula: Seminario de Dpto. de Electrónica y Computadores

Objetivo: Conocer los recursos que ofrece UML para modelar y describir componentes, aplicaciones y sistemas concurrentes y distribuidos. Utilizar las herramientas UML CASE para el desarrollo aplicaciones en Java.

Práctica 1,2,3: Implementación aplicaciones utilizando Java concurrente. (6 horas)

Lab. de Tiempo Real. (Dpto. de Electrónica).

Objetivo: Transformación de una aplicación secuencial a una aplicación concurrente. Implementación de la aplicación utilizando diferentes patrones de concurrencia.

Práctica 4: Implementación de una aplicación distribuida utilizando sockets. (2 horas)

Laboratorio de Tiempo Real (Dpto. de Electrónica).

Objetivo: Implementación de una aplicación concurrente como aplicación distribuida utilizando sockets.

Práctica 5,6,7: Implementación de una aplicación distribuida utilizando RMI. (6 horas)

Laboratorio de Tiempo Real (Dpto. de Electrónica).

Objetivo: Transformación de una aplicación concurrente a aplicación distribuida utilizando Java RMI.

Proyecto 8: Diseño de una aplicación distribuida. (1 horas)

Seminario (Dpto. de Electrónica).

Objetivo: Diseño de una aplicación distribuida a partir de su especificación.

Práctica 9,10: Implementación de una aplicación distribuida utilizando ICE. (4 horas)

Laboratorio de Tiempo Real (Dpto. de Electrónica).

Objetivo: Transformación de una aplicación concurrente a una aplicación distribuida implementada utilizando el middleware ICE.

PROGRAMACIÓN SEMANAL

Lunes y Martes (8:30-9:30): Avance conceptual

Viernes 8:30-9:30: Análisis de un ejemplo. Y trabajo personal siguiente semana.

Miércoles 8:30-10:30 ó Jueves 8:30-10:30: Trabajo en laboratorio.

LIBROS DE TEXTO

DRAKE J.M.: "Programación concurrente y distribuida".

[http://www.ctr.unican.es/asignaturas/procodis_3_II/index.html]

Básica

- JAVADOC, referencia de la Api J2SE 6.0. Disponible online.
- 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)
- Michi Henning, Mark Spruiell: Distributed Programming with Ice (ZeroC).
Disponible online
- Booch et al: El Lenguaje Unificado de Modelado. Guía del Usuario/, 2da edic. Pearson Educación (2006)

Complementaria

- George Coulouris, Jean Dollimore, Tim Kindberg: "Distributed Systems: Concepts and Design" Addison Wesley, 4 edition (2005)
- Charles M. Kozierok: The TCP/IP guide : a comprehensive, illustrated internet protocols reference (2005)
- RFC 793 (TCP), RFC 791(IP), RFC 768 (UDP). Disponible online.
- 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)
- Robert Orfali, Dan Harkey, and Jeri Edwar: "The esencial Distributed Objects Survival Guide" John Wiley & Sons (1999)

- Dan Harkey, Robert Orfali :“Client/Server Programming with Java and CORBA” John Wiley & Sons; 2nd Bk&Cdr edition (1998)
- Andrew Wellings: Concurrent and Real-Time Programming in Java (2004)
- Marko Boger: “ Java in Distributed Systems: Concurrency, Distribution and Persistence” John Wiley & Sons, 1 edition (2001)
- Miles y Hamilton :Learning UML 2.0. O’Reilly (2006)

SISTEMA DE EVALUACION:

Al alumno se le evaluará de forma continuada a través del trabajo que realice individualmente en el laboratorio, así como de las memorias que se le encargue realizar sobre las aplicaciones que haya desarrollado. Esta actividad se evalúa con los siguientes criterios:

- Aprobado; Demuestra conocer los conceptos y manejar las herramientas.
- Notable: Contribuye de forma creativa al desarrollo de la aplicación.
- Sobresaliente y MH. Demuestra dominar el proceso de desarrollo de aplicaciones concurrentes y distribuidas demostrando gran iniciativa y creatividad en el desarrollo de las mismas.

La evaluación continuada se evaluará entre 0 y 8 puntos. Se darán dos calificaciones parciales:

- 16-11-11 y 17-11-11: Programación concurrente
- 18-01-11 y 19-01-11: Programación distribuida

Si el alumno está conforme con la calificación parcial esta pasa a ser la calificación definitiva.

Si el alumno no se encuentra conforme con la calificación que se le asigna en el proceso de evaluación continuada, podrá presentarse al examen final que se realizará en el Laboratorio y será evaluado entre 0 y 10. En este caso la calificación será 0.5 por la calificación continuada más 0.6 por la calificación del examen final.

FECHAS DE EXÁMENES:

Evaluaciones continuadas:

Programación concurrente: 16 y 17 Noviembre
Programación distribuida: 18 y 19 de Enero

Exámenes Finales (Fijado por Facultad):

Febrero: día: 31-01-12 hora: 9:00
Septiembre: día: 05-09-121 hora: 16:00