



Programación Concurrente / Programa de Asignatura

Año Académico 2011- 2012

Titulación	Ingeniero en Informática / Ing. Técnico en Informática de Sistemas
Curso	3º
Créditos	3 de teoría + 3 de práctica
Carácter	Obligatoria
Temporalidad	2º Cuatrimestre
Profesores	Fernando Sánchez Figueroa (fernando@unex.es)

Objetivos / Competencias

Objetivo General:

- Estudiar los fundamentos de la programación concurrente.

Competencias:

- Entender cuándo un problema es de naturaleza concurrente.
- Conocer e identificar los principales problemas que se presentan en programación concurrente.
- Conocer las principales primitivas de sincronización tanto en memoria compartida como distribuida.
- Dado un problema de naturaleza concurrente, decidir qué tipo de primitiva es más apropiada
- Resolver problemas de naturaleza concurrente.

Programa Teórico

TEMA 1 CONCEPTOS FUNDAMENTALES

- 1.1 Introducción
- 1.2 Concepto de programación concurrente
 - Programa y proceso
 - Concurrencia
 - Programación concurrente
- 1.3 Beneficios de la programación concurrente
 - Velocidad de ejecución
 - Solución de problemas inherentemente concurrentes
 - Mejor aprovechamiento de la CPU
- 1.4 Concurrencia y arquitecturas hardware

- Sistemas monoprocesador
- Sistemas multiprocesador
- 1.5 Especificación de ejecución concurrente
 - ¿Qué se puede ejecutar concurrentemente?
 - Cómo especificar la ejecución concurrente
- 1.6 Características de los sistemas concurrentes
 - Orden de ejecución de los sistemas concurrentes
 - Indeterminismo
- 1.7 Problemas inherentes a la programación concurrente
 - Exclusión mutua
 - Condiciones de sincronización
- 1.8 Corrección de programas concurrentes
 - Propiedades de seguridad
 - Propiedades de viveza

TEMA 2 PROBLEMAS DE SINCRONIZACIÓN. PRIMERAS SOLUCIONES

- 2.1 Tipos de sincronización
 - 2.1.1 Condición de sincronización
 - 2.2.2 Exclusión mutua
- 2.2 Solución a la condición de sincronización
- 2.3 Soluciones a la exclusión mutua
 - 2.3.1 Soluciones Software
 - Primeros intentos
 - Algoritmo de Peterson
 - Algoritmo de Dekker
 - Solución para N procesos: Algoritmo de Lamport
 - 2.3.2 Soluciones Hardware
 - Instrucción TestAndSet
 - 2.3.3 Otras soluciones: deshabilitación de interrupciones

TEMA 3 SEMÁFOROS

- 3.1. Introducción
- 3.2. Definición de semáforo
- 3.3. Resolución de problemas usando semáforos
 - 3.3.1 Exclusión mutua y condición de sincronización
 - 3.3.2 Problemas típicos:
- 3.4. Implementación de semáforos
 - 3.4.1 Diferentes posibilidades
 - 3.4.2 Implementación en Java
- 3.5. Inconvenientes de semáforos

TEMA 4 MONITORES

- 4.1 Introducción
- 4.2 Definición de monitor
- 4.3 Condición de sincronización en monitores
- 4.4 Resolución de problemas usando monitores
- 4.5 Equivalencia entre monitores y semáforos
- 4.6 Semántica de la operación resume
- 4.7 Implementación de monitores en Java

TEMA 5 MECANISMOS DE PASO DE MENSAJE

- 5.1 Introducción
- 5.2 Identificación en el proceso de comunicación
- 5.3 Sincronización
- 5.4 Canal de comunicación y mensajes
- 5.5 Espera selectiva

TEMA 6 PASO DE MENSAJE ASÍNCRONO

- 6.1 Introducción
- 6.2 Resolución de problemas con paso de mensaje asíncrono
- 6.3 Implementación del paso de mensaje asíncrono en Java

TEMA 7 PASO DE MENSAJE SÍNCRONO CON CANALES

- 7.1 Introducción
- 7.2 Espera selectiva
- 7.3 Implementación en Java
- 7.4 Resolución de problemas con canales

TEMA 8 INVOCACIÓN REMOTA Y RPC

- 8.1 Introducción
- 8.2 Invocación remota
 - 8.2.1 Implementación en Java
 - 8.2.2 Resolución de problemas con canales
 - Productor/Consumidor
 - Lectores/Escritores
 - Filósofos

TEMA 9 MULTITHREADING EN JAVA

- 9.1 Procesos vs. Hilos
 - Procesos
 - Hilos
 - Estándares
 - Implementación
 - Planificación
- 9.2 Creación de threads en Java
- 9.3 Sincronización de threads en Java
- 9.4 Resolución de problemas

TEMA 10 INTERBLOQUEOS

- 10.1 Definición del interbloqueo
- 10.2 Caracterización del interbloqueo
- 10.3 Tratamiento del interbloqueo

TEMA 11 MODELADO DE SISTEMAS CONCURRENTES

- 11.1 Introducción a las Redes de Petri (RdP)
- 11.2 Modelado de problemas con RdP

Contenido práctico

Se impartirán 2 horas semanales de prácticas en una sola sesión donde se plantearán y resolverán problemas relacionados con la sincronización de procesos utilizando distintas primitivas. Las clases se estructurarán de la siguiente forma:

- una primera hora donde el profesor soluciona un caso práctico.
- una segunda hora donde el profesor propone un caso práctico a resolver por los alumnos. El profesor facilitará un esqueleto de la solución

Metodologías / Actividades

La asignatura de Programación Concurrente busca la participación activa de los estudiantes por lo que se hará un uso intensivo del Campus Virtual.

En las clases de teoría se combinan las clases expositivas con la resolución de problemas.

En cuanto a las clases de laboratorio, estará a disposición de los alumnos, antes de cada sesión un guión, que contendrá los objetivos y los trabajos que se deben desarrollar. Durante el desarrollo de la sesión se utilizará dicho guión.

Medios materiales utilizados:

Pizarra - Cañón de video - Ordenador – Internet

Materiales y recursos utilizados:

Los materiales y recursos utilizados estarán disponibles en el espacio reservado para la asignatura en el Campus Virtual. Concretamente los alumnos dispondrán de:

- Transparencias para cada tema del programa
- Guiones de las sesiones de laboratorio
- Recopilación de exámenes de otros años
- Agenda del curso Además se utilizarán los siguientes recursos:
- Foros de preguntas y respuestas
- Tablón de anuncios de novedades
- Glosarios de términos, palabras claves
- Baterías de preguntas de test
- Conjunto de referencias web relacionadas con la programación concurrente
- Tests de autoevaluación de contenidos
- Tareas virtuales para la entrega de problemas propuestos

Recomendaciones

Se recomienda tener aprobada la asignatura de Estructuras de Datos y Algoritmos y Laboratorio de Programación II.

Criterios evaluación

La materia se divide en dos partes, una parte Teórica y otra Práctica. Para aprobar el curso completo ha de obtenerse una nota mínima de 5.

Teoría:

- Se realizará un examen final en el mes de Junio consistente en un ejercicio tipo test y otro ejercicio de resolución de casos prácticos.

Prácticas:

- Se realizarán varias prácticas a lo largo del cuatrimestre, relacionadas con la parte teórica.
- El número, tipo y evaluación de las prácticas dependerá de la disponibilidad de las salas.
- A final de curso habrá que presentar la implementación de un caso práctico propuesto por el profesor.

Bibliografía

BIBLIOGRAFÍA	
Bibliografía Básica	
[Ben82]	M. Ben-Ari. Principles of concurrent programming. Prentice-Hall
[Bur93]	A. Burns, G.L. Davies. Concurrent Programming. Addison-Wesley

[Dav92a]	G.L. Davies. Pascal-FC, versión 5. User Guide for Pc compatibles. Universidad de Bradford, UK
[Dav92b]	G.L. Davies. Pascal-FC, versión 5. Language Reference Manual. Universidad de Bradford., UK
[Dei98]	H.M. Deitel, P.J. Deitel. Cómo programar en Java. Prentice Hall
[Har98]	S. J. Hartley. Concurrent Programming. The Java Programming Language. Oxford University Press.
[Mag99]	J. Magee, J. Kramer. Concurrency. State model & Java Programs. John Wiley & Sons
[Oak97]	S. Oaks, H. Wong. Java threads. O'Reilly & Associates
[Pal03]	J.T. Palma, M.C. Garrido, F. Sánchez, A. Quesada. Programación Concurrente. Editorial Thomson-Paraninfo (www.paraninfo.es)
[Per90]	J.E. Pérez. Programación Concurrente. Editorial Rueda.
Bibliografía Complementaria	
[And91]	G.R. Andrews. Concurrent Programming. Principle and Practice. Addison-Wesley
[Axf89]	T. Axford. Concurrent Programming. Fundamental Techniques for Real-Time and Parallel Software Design. Editorial Wiley. Series en Parallel Computing
[Bac98]	J. Bacon. Concurrent Systems. Operating Systems, Database and Distributed Systems: An Integrated Approach. Addison Wesley
[Bar98]	J. Barnes. Programming in Ada95. Addison-Wesley
[Ben90]	M. Ben-Ari. Principles of Concurrent and Distributed Programming. Prentice-Hall. Englewood Cliffs N.J.
[Bev97]	J. Beveridge, R. Wiener. Multithreading Applications in Win32. Addison-Wesley
[Bus88]	D. Bustard, J. Elder, J. Welsh. Concurrent Program Structures. Prentice Hall International Series in Computer Science
[But97]	D. R. Butenhof. Programming with POSIX Threads. Addison-Wesley Professional Computing Series
[Dea00]	D. Lea. Programación Concurrente en Java. Principios y patrones de diseño. Addison Wesley
[Dei90]	H.M. Deitel. Sistemas Operativos. Addison-Wesley Iberoamericana, 2ª edición
[Geh88]	N. Gehani, A.D. McGettrick. Concurrent Programming. International Computer Science Series. Addison-Wesley
[Hol00]	Allen Holub. Taming Java Threads. Apress
[Hyd99]	Paul Hyde. Java Thread Programming. SAMS (división de MacMillan Computer Publishing)
[Jon88]	G. Jones, M. Goldsmith. Programming in Occam 2. Prentice Hall. International Series in Computer Science
[Lew00]	B. Lewis, D. J. Berg. Multithreaded programming with Java Technology. Sun Microsystems Press

[Mil94]	M. Milenkovic. Sistemas Operativos. Concepto y diseño. McGraw-Hill, 2ª edición
[Nic96]	B. Nichols, D. Buttler, J. Proulx. Pthreads Programming. O'Reilly
[Pap95]	M. Papathomas. Concurrency in Object-Oriented Programming Languages. En Object Oriented Software Composition, O. Nierstrasz y D. Tschritzis, eds., Prentice-Hall.
[Pet93]	J.L. Peterson, Abraham Silberschatz. Sistemas Operativos. Editorial Reverté
[Sil91]	A. Silberschatz, J. Peterson, P. Galvin. Sistemas Operativos. Conceptos fundamentales. Addison-Wesley Iberoamericana
[Sil85]	M. Silva. Las redes de Petri en la Automática y en la Informática. Editorial AC.
[Tan92]	A. S. Tanenbaum. Sistemas Operativos modernos. Prentice Hall

Horas de estudio recomendadas

Se recomienda una dedicación no presencial de aproximadamente 3 horas de media a la semana.

Normas

- La convocatoria del Examen de Teoría será fijada por la Subdirección Académica del Centro. El profesor sólo fijará la hora de comienzo del examen teórico y las pruebas prácticas.
- De acuerdo a la normativa del Centro, para aquellos alumnos que la fecha del examen oficial coincida con la fecha del examen de otra asignatura, deberán solicitar el cambio de fecha del examen mediante instancia presentada en la secretaría del centro, en el período establecido por la misma. NO se realizará ningún examen extra si no se ha seguido el procedimiento oficial establecido.
- Los alumnos rellenarán sus datos en el Aula Virtual (especialmente aportarán una foto reciente y clara) durante las dos primeras semanas de clase.