
Programación Concurrente

Joaquín EZPELETA
Dept. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza



Programas concurrentes

- Un programa concurrente se compone de *procesos y objetos compartidos*
- Un proceso es un *programa secuencial* ejecutado en algún procesador
- Los objetos compartidos se construyen o bien mediante *memoria compartida* o bien mediante una *red de comunicación*
- Un programa concurrente se puede ejecutar de varias maneras:
 - un único procesador
 - sistemas multi-procesador

multiprogramación

• *multiprocesamiento*
• *programación distribuída*

Programas concurrentes

- Sobre sistemas multi-procesador:
 - permite paralelismo real
 - sistemas fuertemente acoplados
 - todos los elementos (incluida la memoria compartida) conectados a un mismo bus
 - comunicación y sincronización en memoria compartida (al margen de la existencia de memoria local a cada procesador)
 - sistemas débilmente acoplados
 - hay comunicación entre procesadores, pero no memoria compartida
 - habitualmente, comunicación mediante paso de mensajes

Programas concurrentes

- A modo de clasificación:
 - un **programa concurrente** contiene acciones que se pueden ejecutar simultáneamente.
 - un **programa paralelo** es un programa concurrente diseñado para ejecutarse en un *hardware* paralelo.
 - un **programa distribuido** es un programa paralelo diseñado para ejecutarse en una red de procesadores autónomos que no comparten memoria.

Programas concurrentes

- Para cooperar, los procesos deben comunicar
 - la comunicación permite a un proceso influir en la ejecución de otro
- La comunicación puede tener retrasos
 - lo que implica que la información obtenida por un proceso respecto a otro puede estar desfasada
- En consecuencia: **diseñar y verificar programas concurrentes correctos es más difícil que diseñar un conjunto de procesos correctos**

Objetivos

- Estudiar, desde un punto de vista formal, los conceptos fundamentales ligados a la programación concurrente
 - “Re-visitación”, desde una perspectiva formal, conceptos estudiados en asignaturas de SO
- Aprender los mecanismos y técnicas de sincronización fundamentales
- Aprender a razonar sobre la corrección de programas concurrentes
- Practicar la implementación de programas concurrentes en Ada

La asignatura

- Material:
 - <http://www.cps.unizar.es/~ezpeleta/>
- Valor de la asignatura: $3.0 + (1.5 + 1.5) = (3.0 + 1.5) + 1.5$
- Evaluación:
 - Teoría: hasta 4.0 puntos (≥ 2.0)
 - Prácticas: hasta 2.0 puntos (≥ 1.5)
 - Trabajo fin de curso: hasta 4.0 puntos (≥ 2.0)
- Profesorado:
 - Joaquín Ezpeleta
 - ezpeleta@unizar.es
 - Miguel Ángel Latre
 - latre@unizar.es

La asignatura

- Bibliografía básica:
 - Andrews G.R.
Concurrent Programming. Principles and Practice
Benjamin Cummings Company, 1991
 - Burns A. Welling A.
Concurrency in Ada
Cambridge University Press, 1995
- Bibliografía complementaria:
 - Ben-Ari M.
Principles of Concurrent and Distributed Programming
Addison-Wesley, 2006
 - Snow C.R.
Concurrent Programming
Cambridge Computer Science Texts, 1992
 - Schneider F.B.
On Concurrent Programming
Graduate Texts in Computer Science, Springer-Verlag, 1997
 - J.T. Palma, M.C. Garrido, F. Sánchez, A. Quesada
Programación Concurrente
Thompson, 2003
 - M.L. Liu
Computación distribuida. Fundamentos y aplicaciones
Ed. Pearson- Addison Wesley, 2004.

Contenidos del curso

Lección 1: Lenguaje algorítmico para programas secuenciales

Lección 2: Conceptos básicos de PC

Lección 3: Concurrencia y sincronización

Lección 4: Semántica de la composición concurrente

Lección 5: Técnicas para asegurar ausencia de interferencias

Lección 6: Propiedades de un programa concurrente

Lección 7: Sincronización de procesos.
El problema de la sección crítica

Lección 8: Sincronización por barrera

Contenidos del curso

Lección 9: Programación concurrente con semáforos

Lección 10: Ejemplos de programación con semáforos

Lección 11: Sincronización mediante monitores

Lección 12: Ejemplos de programación con monitores

Lección 13: Introducción a la programación distribuida

Lección 14: Programación mediante paso asíncrono de mensajes

Lección 15: El modelo de coordinación Linda

Lección 16: Programación mediante paso síncrono de mensajes