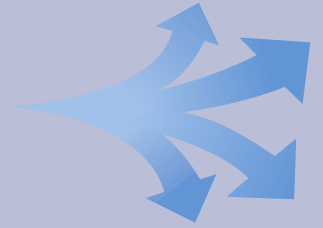


Sistemas Concurrentes y Distribuidos

Grado en Ingeniería Informática



Depto. de Lenguajes y Sistemas Informáticos
Universidad de Granada



Datos de la Asignatura

PÁGINA WEB: <https://pradogrado.ugr.es>

Profesor: José Miguel Mantas Ruiz

jmmantas@ugr.es

<http://lsi.ugr.es/~jmantas/>

Despacho 20, planta 3ª

- **Tutorías:** Lunes (09:30 a 13:30),
Martes (11:30 a 13:30).



Objetivos

- Comprender la **importancia de la programación concurrente** (PC) hoy en día y los problemas que plantea.
- Identificar las características de los distintos **tipos de sistemas concurrentes**.
- Entender los conceptos de **sincronización y exclusión mutua** entre procesos.
- Identificar y analizar **propiedades** de un **sistema concurrente**.
- Conocer los principales **modelos de programación concurrente**.
- Adquirir experiencia y conocimiento en los **mecanismos de sincronización y comunicación** para PC, tanto con memoria compartida, como para sistemas distribuidos.
- Entender y saber usar **semáforos y monitores** para resolver problemas.
- Ser capaz de **desarrollar algoritmos que resuelvan problemas modelo en PC** en los contextos más frecuentes.
- Ser capaz de usar **bibliotecas y plataformas estandarizadas** de PC.
- Conocer las técnicas más destacadas para el **diseño de sistemas de tiempo real**.

Programa de Teoría

- 1. Introducción a la Programación Concurrente.**
- 2. Algoritmos y mecanismos de sincronización basados en memoria compartida.**
- 3. Sistemas basados en paso de mensajes**
- 4. Introducción a los sistemas de tiempo real.**



Tema 1: INTRODUCCIÓN A LA PROGRAMACIÓN CONCURRENTES

- 1.1. Conceptos básicos y Motivación**
- 1.2. Modelo Abstracto y Consideraciones sobre el Hardware**
- 1.3. Notaciones para expresar ejecución concurrente**
- 1.4. Exclusión mutua y Sincronización**
- 1.5. Propiedades de sistemas concurrentes. Nociones de verificación**



Tema 2: Algoritmos y mecanismos de sincronización basados en memoria compartida.

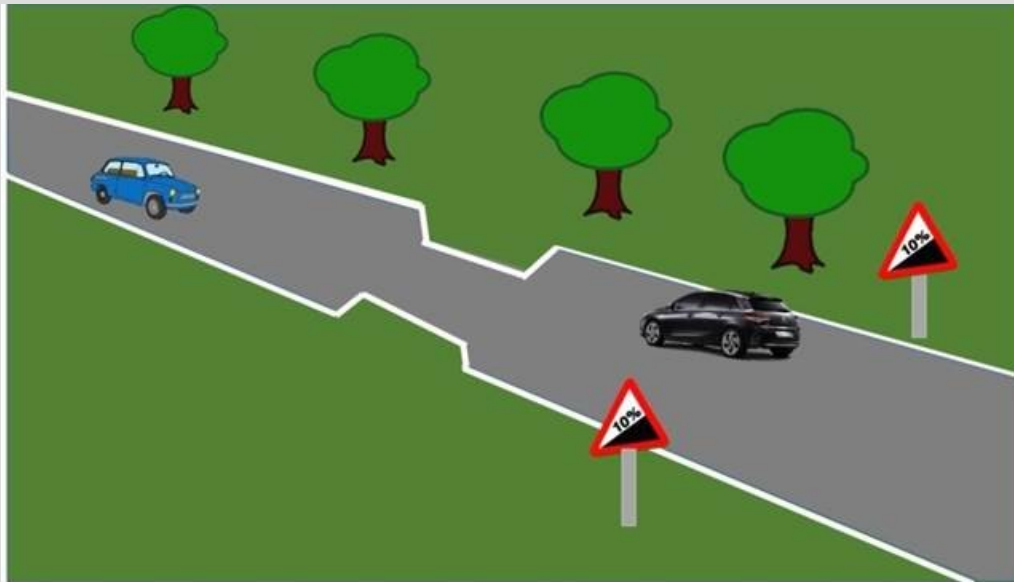
2.1. Introducción a la sincronización en memoria compartida

2.2. Semáforos para sincronización

2.3. Monitores como mecanismo de alto nivel

2.4. Soluciones software con espera ocupada para Exclusión Mutua (EM)

2.5. Soluciones hardware con espera ocupada (cerrojos) para EM

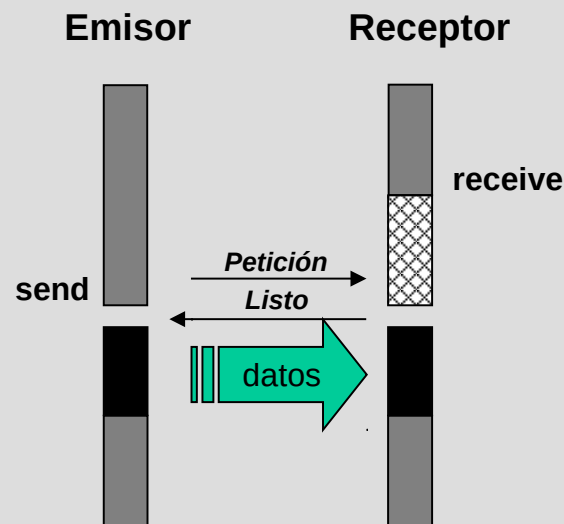


Tema 3: SISTEMAS BASADOS EN PASO DE MENSAJES

3.1: Mecanismos básicos en paso de mensajes.

3.2: Patrones de Interacción en programas paralelos y distribuidos

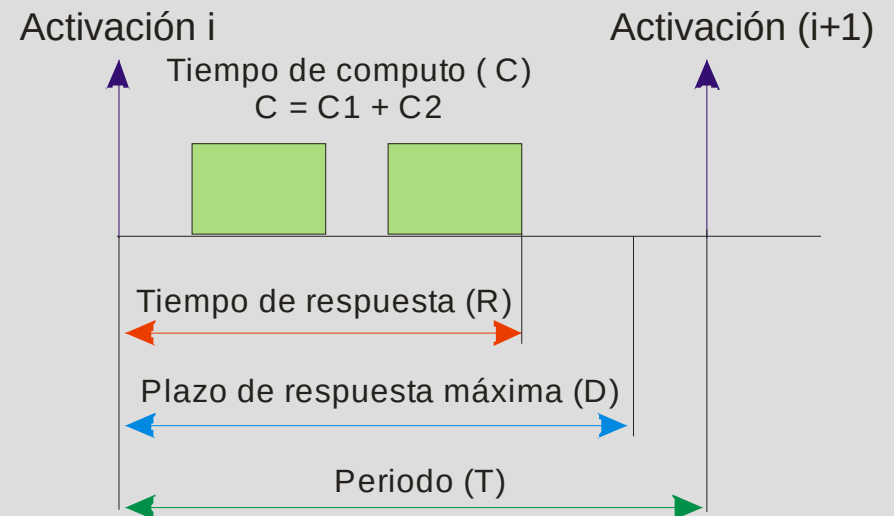
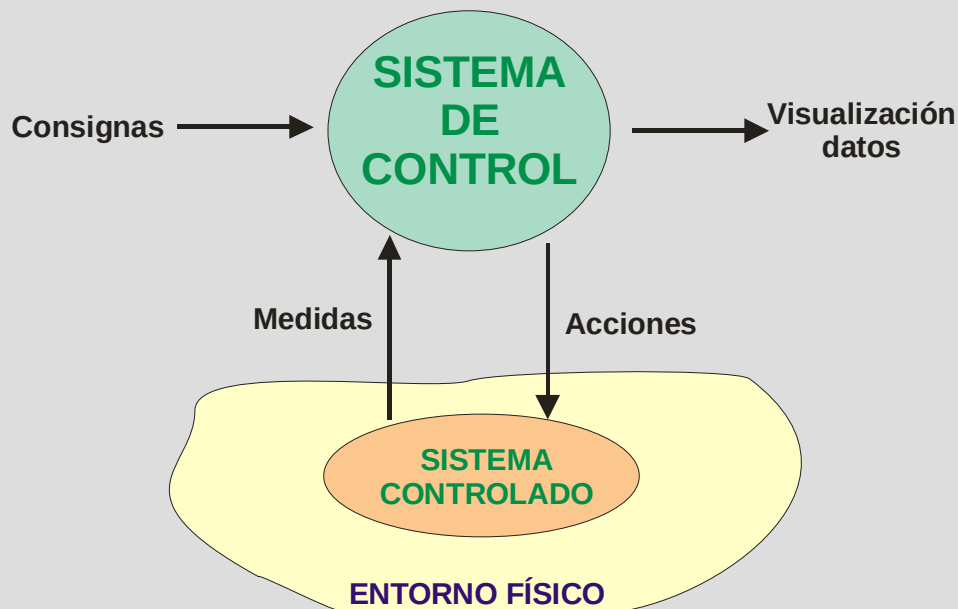
3.3: Mecanismos de alto nivel para sistemas distribuidos.



Tema 4: Introducción a los Sistemas de Tiempo Real

4.1. Concepto de sistema de tiempo real. Medidas de tiempo y modelo de tareas.

4.2. Planificación de tareas periódicas con asignación de prioridades.



Programa de Prácticas

- 1. Resolución de problemas de sincronización con semáforos.**
- 2. Programación de monitores con hebras.**
- 3. Programación de aplicaciones distribuidas.**
- 4. Programación de tareas periódicas con prioridades.**

Seminarios

- 1. Introducción a la programación mutihebra usando semáforos.**
- 2. Introducción a la programación multihebra con monitores.**
- 3. Introducción al uso de una interfaz de paso de mensajes.**

Bibliografía Fundamental

- J. T. Palma, C. Garrido, F. Sánchez, A. Quesada. **Programación Concurrente** Paraninfo. 2008. 2ª edición.
- G. R. Andrews. **Foundations of Multithreaded, Parallel, and Distributed Programming**. Addison Wesley, 2000.
- Francisco. Almeida, Domingo Giménez, José M. Mantas, Antonio Vidal. **Introducción a la Programación Paralela**. Paraninfo, 2008.
- V. Kumar , A. Grama, A. Gupta, G. Karypis. ***Introduction to Parallel Computing***. Benjamin/Cummings Publishing Company, 2003. Anthony Williams.
- ***C++ Concurrency in Action, 2nd Edition***. Manning Publications. 2018.
- George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. ***Distributed Systems: Concepts and Design***. Addison-Wesley, 2011.
- A. Burns, A. Wellings. ***Sistemas de Tiempo Real y Lenguajes de Programación (3ª Edición)***. Addison Wesley, 2003.

Metodología Docente

Actividades de enseñanza-aprendizaje :

- 1) **Actividades presenciales:** Clases magistrales, Resolución de ejercicios/problemas individuales y/o en grupo, Sesiones prácticas en laboratorio, Tutorías y Pruebas objetivas
- 2) **Actividades no presenciales:** Estudio individual o en grupo, Realización de ejercicios/problemas/trabajos, tanto individuales como en grupo, Confección de la carpeta de aprendizaje o portafolio de prácticas.

Evaluación → doble misión en el aprendizaje:

- **Adquisición de competencias** (ver apartado siguiente),
- **Evaluación formativa:** Se fomenta la **continua retroalimentación** del estudiante, que puede conocer en todo momento cómo progresa. Para ello, se resolverán las pruebas objetivas, ejercicios, problemas, etc.

Evaluación (1)

Cálculo calificación final de la nota

- 65% de teoría
- 35% prácticas.

Para aprobar:

- Calificación numérica: al menos **5 (sobre 10)**.
- Tanto calificación parte teórica como práctica: al menos 40% máxima calificación.

Por defecto: **Evaluación continua:**

- **Asistir a clase regularmente** (no será obligatoria, salvo las sesiones en las que se programen pruebas de evaluación) y realizar las pruebas y ejercicios que se plantean
- **Asistir a todas las pruebas objetivas de teoría y prácticas**, aunque se admite que, de forma justificada, se falte a una prueba como máximo.

Evaluación(2)

Calificación Teoría

La calificación de teoría (**6,5 puntos**) se reparte entre las pruebas objetivas individuales realizadas al final de ciertos temas.

- ♦ **Prueba temas 1 y 2:** durante el periodo de clases.
- ♦ **Prueba temas 3 y 4:** en la fecha fijada para el examen final.

Distribución de la puntuación máxima por temas:

- **Temas 1 y 2:** 3,5
- **Temas 3 y 4:** 3

- Hasta **0,3 puntos extra** (con límite máximo de 6,5 puntos para parte teórica), correspondientes a **resolución voluntaria de ejercicios, problemas, y/o trabajos**.

Evaluación(3)

Calificación Prácticas

La calificación de prácticas (**3,5 puntos**) se reparte entre las 4 pruebas objetivas que se realizan durante la última sesión de prácticas de la parte a evaluar.

Distribución de la puntuación máxima por prácticas:

- Pract. 1: 0,8
- Pract. 2 : 1,2
- Pract. 3: 1,2
- Pract. 4: 0,3

- Hasta **0,2 puntos extra** (con límite máximo de 3,5 puntos para la parte práctica), correspondientes a las **soluciones de los ejercicios propuestos en seminarios**.

Evaluación(4)

Evaluación Única final

De acuerdo a la Normativa de evaluación y de calificación de los estudiantes de la Universidad de Granada, aquellos alumnos que tengan **algún motivo justificado para no poder seguir la modalidad de evaluación continua**, podrán **solicitar al director del Departamento de Lenguajes y Sistemas Informáticos** que la evaluación sea por la modalidad de evaluación única final.

Dos pruebas de evaluación (parte teórica y parte práctica):

- **Evaluación de teoría:** Prueba escrita con preguntas de teoría, tipo test y problemas.
- **Evaluación de prácticas:** Prueba en el laboratorio con ejercicios de programación.

Para aprobar:

- ✓ Nota prueba teoría es igual o superior al 40% del máximo.
- ✓ Nota prueba de prácticas es igual o superior al 40% del máximo.
- ✓ La suma ponderada (65% parte teórica y 35% parte práctica) debe ser al menos el 50% del máximo posible.

Evaluación(5)

En el caso de que el alumno no apruebe la asignatura, pero tenga una **nota igual o superior al 40% en alguna de las dos partes** (teoría o práctica), **se podrá guardar** dicha nota para las **convocatorias extraordinarias del presente curso**. En ambas convocatorias se aplicará el mismo modelo que en la evaluación única final.

En el caso de que algún alumno no apruebe la asignatura, pero tenga **una nota igual o superior al 50% (1,75 puntos)** en la parte **práctica**, obtenida mediante la modalidad de evaluación continua, **se podrá guardar dicha nota para todas las convocatorias de cursos posteriores**