

**12745 - PROGRAMACIÓN
CONCURRENTE**

ASIGNATURA: 12745 - PROGRAMACIÓN CONCURRENTE

Vinculado a : (Titulación - Asignatura - Especialidad)

1801-Ingeniería en Informática - 12745-PROGRAMACIÓN CONCURRENTE - 00

CENTRO: Escuela de Ingeniería Informática

TITULACIÓN: Ingeniero en Informática

DEPARTAMENTO: INFORMÁTICA Y SISTEMAS

ÁREA: Ciencia De La Comp. E Intel. Artificial

PLAN: 10 - Año 199**ESPECIALIDAD:**

CURSO: Cr. comunes cic**IMPARTIDA:** Segundo semestre **TIPO:** Optativa

CRÉDITOS: 6

TEÓRICOS: 3

PRÁCTICOS: 3

Información ECTS

Créditos ECTS: 4,8

Horas de trabajo del alumno: 120

Horas presenciales:

- Horas teóricas (HT): 30
- Horas prácticas (HP): 30
- Horas de clases tutorizadas (HCT): 6
- Horas de evaluación: 6
- otras:

Horas no presenciales:

- trabajos tutorizados (HTT): 19
- actividad independiente (HAI): 29

Idioma en que se imparte: Español

Descriptores B.O.E.

Bloqueos. Exclusión Mutua. Sincronizaciones y Comunicaciones. Esquemas de Competencia y Cooperación. Semáforos. Buffers. Gestión de Regiones Críticas. Monitores. Lenguajes de Programación.

Temario

1. Fundamentos de la concurrencia (5h) [1]
 - 1.1.- Concepto de programación concurrente
 - 1.2.- Beneficios de la programación concurrente
 - 1.3.- Concurrencia y arquitecturas hardware
 - 1.4.- Especificación de ejecución concurrente
 - 1.5.- Características de los sistemas concurrentes
 - 1.6.- Problemas inherentes a la programación concurrente
 - 1.7.- Corrección de programas concurrentes

2. Sincronización y comunicación con memoria compartida (10h) [1]
 - 2.1.- El problema de la sección crítica.
 - 2.2.- Primeras aproximaciones a la solución de los problemas de la programación concurrente
 - 2.2.- Semáforos.
 - 2.3.- Regiones críticas
 - 2.4.- Monitores
3. Sincronización y comunicación con paso de mensajes (15h) [1]
 - 3.1.- Mecanismo de paso de mensaje
 - 3.2.- Paso de mensaje asíncrono
 - 3.3.- Paso de mensaje síncrono con canales
 - 3.4.- Invocación remota y llamada a procedimiento remoto

Requisitos Previos

- Programación estructurada, tipos abstractos de datos, verificación de algoritmos (Estructuras de Datos I y II, Metodología de la Programación, Tecnología de la Programación).
- Sistemas operativos (asignatura del mismo nombre), especialmente fundamentos sobre procesos concurrentes, planificación y sincronización.
- Operación en entorno Linux: uso del shell, editores, entorno de ventanas, etc.

Objetivos

Caben varios enfoques en una asignatura de programación concurrente. Uno de ellos, utilizado en bastantes universidades, es partir de un enfoque formal y matemático, estudiando métodos y notaciones que permiten probar la corrección de los programas. Este enfoque requeriría una formación previa en matemática discreta y verificación de algoritmos que no parece encajar en el perfil curricular del actual plan de estudios. Aquí se propone una aproximación algo más práctica, en la que los fundamentos teóricos se expliquen de una forma más intuitiva que rigurosa, aunque por supuesto concediéndoles la importancia que merecen.

En cuanto a las herramientas, tenemos que hoy día existen distintas familias de primitivas de programación concurrente (semáforos, monitores, objetos protegidos, paso de mensajes...), sin que ninguna haya prevalecido claramente: el estudiante tendrá que aprender todos los grandes modelos.

Otro aspecto a considerar es el lenguaje de programación a utilizar en la presentación de los contenidos propios de la programación concurrente. Lo primero que hay que tener en cuenta es que no existe un lenguaje, que sea utilizado en aplicaciones reales, que proporcione todos los mecanismos utilizados en programación concurrente. Por tanto, sería necesario introducir varios lenguajes a lo largo del cuatrimestre para poder realizar prácticas con cada uno de ellos, lo cual no es abordable en el tiempo disponible. Debido a esto, se ha optado por utilizar un lenguaje especialmente diseñado para docencia en programación concurrente, el Pascal-FC. El Pascal-FC proporciona un conjunto bastante amplio de primitivas que pueden ser utilizadas de forma conjunta, y que nos permiten presentar de una manera clara y sencilla cada uno de los tópicos tratados en las clases teóricas. Sin embargo, es importante analizar un lenguaje que permita a los alumnos resolver problemas reales. Para cubrir este hueco, los alumnos estudiarán y desarrollarán ejemplos con Java y/o Ada. Además, en la asignatura se explicarán otros lenguajes con soporte para concurrencia, como Occam y SR.

En consecuencia, los objetivos de formación de esta asignatura serán:

- (1) Conocimiento básico de los fundamentos de la programación concurrente
- (2) Capacidad para construir programas concurrentes
- (3) Capacidad para analizar programas concurrentes

Metodología

Se asumirá que los estudiantes han tenido cierto contacto con la programación concurrente, en la asignatura de Sistemas Operativos, especialmente en el modelo de memoria compartida.

Los conceptos teóricos serán explicados mediante clases magistrales y resolución de ejercicios en pizarra u ordenador. Se intentará que los estudiantes estén en disposición de recibir cada concepto de forma razonada y crítica, incluso descubriendo antes por sí mismo lo que se va a explicar. También se encargará a los estudiantes analizar algunos problemas clásicos de concurrencia que tendrán que exponer en clase.

Se tratará también de aplicar todos los conceptos teóricos usando un lenguaje de programación (Pascal-FC, Ada, Java). Los ejercicios y trabajos se realizarán en grupos de dos o tres personas. La mayoría de los ejercicios serán ejercicios de concurrencia básicos, comunes a todos los estudiantes.

Los grupos de trabajo realizarán un trabajo final, que tratará de profundizar en alguna herramienta de programación concurrente o de construir una pequeña aplicación. Este trabajo final habrá de ser expuesto en clase ante los compañeros y el profesor.

Para la realización del software, se emplearán herramientas de programación gratuitas y de fácil obtención: Linux, compilador Pascal-FC, compilador GNAT, JDK, etc., con el doble fin de evitar inversiones innecesarias por parte del Departamento y de facilitar que el estudiante pueda realizar sus trabajos fuera de la universidad.

Criterios de Evaluación

La asignatura se calificará según los resultados obtenidos en:

- los trabajos prácticos y exposiciones (40% de la nota total)
- un examen global (40%)
- la participación activa en la asignatura (20%)

En cualquier caso, será obligatorio realizar tanto el examen como los trabajos, amén de obtener una nota mínima en cada uno de ellos. Esta nota mínima es el 35% de la nota máxima de la prueba. En otro caso, la calificación de la asignatura será de suspenso.

Descripción de las Prácticas

Práctica número 1

Descripción: Programación en Pascal-FC/Java/Ada

Objetivos: Adiestrarse en la programación elemental y concurrente en Pascal-FC

Material de Laboratorio recomendado: Por cada persona, un pecé con Linux/Windows, compilador Pascal-FC/Java/Ada, entorno gráfico y conexión a la red.

Nº horas estimadas en laboratorio: 2

Práctica número 2

Descripción: Semáforos en Pascal-FC/Java/Ada

Objetivos: Iniciarse en la programación concurrente con Pascal-FC. Resolver ejercicios de sincronización mediante semáforos.

Material de Laboratorio recomendado: Por cada persona, un pecé con Linux/Windows, compilador Pascal-FC/Java/Ada, entorno gráfico y conexión a la red.

Nº horas estimadas en laboratorio: 6

Práctica número 3

Descripción: Monitores en Pascal-FC/Java/Ada

Objetivos: Adiestrarse en el uso de monitores implementando algunos ejercicios propuestos de concurrencia.

Material de Laboratorio recomendado: Por cada persona, un pecé con Linux/Windows, compilador Pascal-FC/Java/Ada, entorno gráfico y conexión a la red.

Nº horas estimadas en laboratorio: 6

Práctica número 4

Descripción: Comunicación mediante canales en Pascal-FC

Objetivos: Iniciarse en la resolución de problemas mediante mecanismos de paso de mensaje, resolviendo problemas de concurrencia mediante canales: espera selectiva, espera selectiva con guardas, ...

Material de Laboratorio recomendado: Por cada persona, un pecé con Linux/Windows, compilador Pascal-FC, entorno gráfico y conexión a la red.

Nº horas estimadas en laboratorio: 6

Práctica número 5

Descripción: Invocación remota en Pascal-FC/Java/Ada

Objetivos: Conocer el esquema de comunicación mediante invocación remota resolviendo algunos problemas de sincronización

Material de Laboratorio recomendado: Por cada persona, un pecé con Linux/Windows, compilador Pascal-FC/Java/Ada, entorno gráfico y conexión a la red.

Nº horas estimadas en laboratorio: 6

+ trabajos de curso, que varían según la naturaleza del trabajo asignado a cada grupo

Nº horas estimadas en laboratorio: 4

Bibliografía

[1 Básico] Concurrent programming /

Alan Burns, Geoff Davies.

Addison-Wesley,, Wokingham, England : (1993)

0201544172

[2 Básico] Programación concurrente /

José Tomás Palma Méndez, María del Carmen Garrido Carrera, Fernando Sánchez Figueroa, Alexis Quesada Arencibia.

Thomson,, Australia ; [Madrid] : (2003)

8497321847

[3 Básico] Principles of concurrent and distributed programming /

M. Ben-Ari.

Prentice-Hall,, New York : (1990)

013711821X

[4 Recomendado] Concurrency in Ada.

Burns, Alan

Cambridge University Press,, Cambridge : (1998) - (2nd ed.)

0-521-62911-X

[5 Recomendado] The origin of concurrent programming: from semaphores to remote procedure calls /

edited by Per Brinch Hansen.

Springer,, New York [etc.] : (2002)

[6 Recomendado] Programming in ADA 95 /

John Barnes.

Addison-Wesley,, Harlow (England) : (1998) - (2nd ed.)

0-201-34293-6

[7 Recomendado] Concurrent programming in Java TM: Design principles and patterns.

Lea, Douglas

Addison-Wesley,, Reading (Massachusetts) : (2000) - (2nd ed.)

0-201-31009-0

Equipo Docente

FRANCISCO ALEXIS QUESADA ARENCIBIA

(COORDINADOR)

Categoría: PROFESOR CONTRATADO DOCTOR, TIPO 1

Departamento: INFORMÁTICA Y SISTEMAS

Teléfono: 928454572 **Correo Electrónico:** alexis.quesada@ulpgc.es

Resumen en Inglés

The fundamentals about concurrency and concurrent programming are explained in this course.

The approach followed is quite practical: provided students with an intuitive grasp of concurrency and the classic problems found in concurrent programming, presenting the synchronization and communication primitives introduced into languages for their solution. It is described many common language primitives, shared-variable interaction, asynchronous and synchronous message passing, rendezvous, semaphores, and monitors.

As an applied activity, the students will develop programs in Pascal-FC/Ada/Java, that supports all the major primitives which are explained.