

Programación Paralela

Código: 43481

Créditos ECTS: 6

2025/2026

Titulación	Tipo	Curso
Modelling for Science and Engineering	OB	1

Contacto

Nombre: Juan Carlos Moure Lopez

Correo electrónico: juancarlos.moure@uab.cat

Equipo docente

Christian Guzman Ruiz

Idiomas de los grupos

Puede consultar esta información al [final](#) del documento.

Prerrequisitos

No hay prerequisitos

Objetivos y contextualización

El objetivo del módulo es:

- Desarrollar programas en C.
- Identificar las dificultades relacionadas con la programación paralela.
- Aplicar una metodología adecuada para el desarrollo de aplicaciones paralelas.
- Comprender las diferencias de los enfoques de programación en paralelo: memoria compartida, paso de mensajes.
- Determinar el enfoque de programación paralela más conveniente para desarrollar una aplicación particular.
- Desarrollar aplicaciones paralelas.
- Evaluar el rendimiento de la aplicación paralela y recopilar las medidas necesarias para ajustar la aplicación a fin de mejorar su rendimiento.

Al final de este módulo, los estudiantes deben tener conocimientos, métodos y habilidades técnicas suficientes para desarrollar aplicaciones paralelas utilizando un modelo de programación adecuado y evaluar el rendimiento de la aplicación.

Resultados de aprendizaje

1. CA04 (Competencia) Comunicar los resultados derivados del tratamiento de problemas de programación paralela a un público experto
2. CA05 (Competencia) Trabajar en equipos multidisciplinares en la aplicación de la programación en paralelo a la modelización de procesos reales
3. KA03 (Conocimiento) Identificar los entornos de programación que permiten el desarrollo de algoritmos de paralelización
4. KA04 (Conocimiento) Identificar las principales arquitecturas de computadores paralelos y distribuidos
5. KA05 (Conocimiento) Describir el funcionamiento y los parámetros en que se basan las herramientas de análisis del rendimiento de una estructura de programación en paralelo
6. SA04 (Habilidad) Usar el software específico para la programación en paralelo
7. SA05 (Habilidad) Desarrollar la solución paralela a un problema computacional eligiendo las herramientas más adecuadas
8. SA06 (Habilidad) Utilizar las herramientas apropiadas para analizar el rendimiento de una aplicación de programación en paralelo

Contenido

1. Introducción al curso.
2. Programación en C.
3. Introducción a los sistemas de altas prestaciones, sistemas paralelos, multiprocesadores y multicamputadores. Ejecución de aplicaciones paralelas en sistemas de altas prestaciones.
4. Programación OpenMP - Aplicaciones basadas en memoria compartida. Estándar OpenMP (Open multiprocessing).
5. Programación MPI - Aplicaciones basadas en paso de mensajes. Estándar MPI (Message Passing Interface).
6. Análisis de rendimiento - Análisis de prestaciones de sistemas paralelos. Evaluación de prestaciones de sistemas paralelos. Ejemplos de herramientas de evaluación de prestaciones.
7. Programación GPU.

Actividades formativas y Metodología

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Asistencia a clases de teoría y ejercicios prácticos	14	0,56	KA03, KA04, KA05, KA03
Prácticas de laboratorio	24	0,96	CA05, SA04, SA05, SA06, CA05
Tipo: Autónomas			
Diseño desarrollo de ejercicios prácticos	52	2,08	KA03, KA04, KA05, KA03
Estudio	36	1,44	KA03, KA04, KA03

La asignatura se desarrollará en clases teóricas y ejercicios prácticos.

Es recomendable que el alumnado asista a todas las clases de la asignatura con un ordenador portátil con la batería bien cargada.

El planteamiento del trabajo está orientado a promover un aprendizaje activo y desarrollar las competencias de capacidad de organización y planificación, comunicación oral y escrita, trabajo en equipo y razonamiento crítico. La calidad de los ejercicios realizados, de su presentación y de su funcionamiento se valorará especialmente.

La gestión de la docencia de la asignatura se hará a través del Campus Virtual (<https://cv.uab.cat/>), que servirá para poder ver los materiales, gestionar los grupos de prácticas, hacer las entregas correspondientes, ver las notas, comunicarse con los/las profesores/as, etc.

Nota: se reservarán 15 minutos de una clase dentro del calendario establecido por el centro o por la titulación para que el alumnado rellene las encuestas de evaluación de la actuación del profesorado y de evaluación de la asignatura o módulo.

Evaluación

Actividades de evaluación continuada

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Ejercicios prácticos con GPUs	20	6	0,24	CA05, SA04, SA05, SA06
Ejercicios prácticos en C	15	6	0,24	CA05, SA04, SA05, SA06
Ejercicios prácticos en MPI	20	6	0,24	CA05, SA04, SA05, SA06
Ejercicios prácticos en OpenMP	15	4	0,16	CA05, SA04, SA05, SA06
Examen Final	30	2	0,08	CA04, KA03, KA04, KA05

El objetivo del proceso de evaluación es verificar que el alumno ha adquirido los conocimientos y habilidades de la asignatura. Se evaluarán cinco tipos de actividades de forma independiente donde la media ponderada de las cinco actividades dará la calificación final.

1. Programación en C (15%)
2. Programación en OpenMP (15%)
3. Programación en MPI (20%)
4. Programación con GPUs (20%)
5. Examen final individual (30%)

Las evaluaciones de la programación serán realizadas en grupos de 2 personas. Examen final será individual y evaluará la programación en C.

Nota sobre plagios:

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, y de acuerdo con la normativa académica vigente, las irregularidades cometidas por un estudiante que puedan conducir a una variación de la calificación en una actividad evaluable se calificarán con un cero (0). Las actividades de evaluación calificadas de esta forma y por este procedimiento no serán recuperables. Si es necesario superar cualquiera de estas actividades de evaluación para aprobar la asignatura, esta asignatura quedará suspendida directamente, sin oportunidad de recuperarla en el mismo curso. Estas irregularidades incluyen, entre otros:

- la copia total o parcial de una práctica, informe, o cualquier otra actividad de evaluación;
- dejar copiar;
- presentar un trabajo de grupo no hecho íntegramente por los y las miembros del grupo (aplicado a todos los y las miembros, no sólo a los que no han trabajado);
- presentar como propios materiales elaborados por un tercero, aunque sean traducciones o adaptaciones, y en general trabajos con elementos no originales y exclusivos del estudiante;
- el uso no autorizado de la IA (p. ej., Copilot, ChatGPT o equivalentes);
- tener dispositivos de comunicación (como teléfonos móviles, smart watches, bolígrafos con cámara, etc.) accesibles durante las pruebas de evaluación teórico-prácticas individuales (exámenes);
- hablar con compañeros o compañeras durante las pruebas de evaluación teórico-prácticas individuales (exámenes);
- copiar o intentar copiar de otros alumnos durante las pruebas de evaluación teórico-prácticas (exámenes);
- usar o intentar usar escritos relacionados con la materia durante la realización de las pruebas de evaluación teórico-prácticas (exámenes), cuando éstos no hayan sido explícitamente permitidos.

Bibliografía

Parallel Programming : Techniques and Applications using Networked Workstations and Parallel Computers. Barry Wilkinson. Prentice Hall, 1999.

Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering. Ian Foster . Addison Wesley, 1995.

Introduction to Parallel Computing. A. Grama et alter. Addison Wesley, Second Edition, 2003.

Parallel Program Development For Cluster Computing: Methodology, Tools and Integrated Environments. Edited by J. C. Cunha, P. Kacsuk, S. C. Winter. Nova Science Publishers, Inc., 2001.

High Performance Cluster Computing (Vols. 1 y 2), Rajkumar Buyya ed., Prentice Hall, 1999.

Parallel Programming with MPI, Peter Pacheco, Morgan Kauffman, 1996

Using MPI-2, William Gropp, Ewing Lusk and Rajeev Thakur, MIT Press, 1999.

Programming Massively Parallel Processors: A Hands-on Approach. David Kirk and Wen-mei Hwu. ISBN: 978-0-12-381472-2. Published by Elsevier Inc. 2010.

http://www.elsevierdirect.com/morgan_kaufmann/kirk/

Software

C

OpenMP

MPI

OpenACC

gestor de colas (SLURM)

herramientas para evaluación de rendimiento (perf, TAU, nvcc)

conexión remota con laboratorio (e.g. MobaTex, Eclipse, Visual Studio, etc.)

Grupos e idiomas de la asignatura

La información proporcionada es provisional hasta el 30 de noviembre de 2025. A partir de esta fecha, podrá consultar el idioma de cada grupo a través de este [enlace](#). Para acceder a la información, será necesario introducir el CÓDIGO de la asignatura

Nombre	Grupo	Idioma	Semestre	Turno
(PLABm) Prácticas de laboratorio (máster)	1	Inglés	primer cuatrimestre	tarde
(PLABm) Prácticas de laboratorio (máster)	2	Inglés	primer cuatrimestre	tarde
(TEm) Teoría (máster)	1	Inglés	primer cuatrimestre	tarde