

Curso Académico: (2024 / 2025)

Fecha de revisión: 08-05-2024

Departamento asignado a la asignatura: Departamento de Informática

Coordinador/a: GARCIA SANCHEZ, JOSE DANIEL

Tipo: Obligatoria Créditos ECTS : 6.0

Curso : 4 Cuatrimestre : 1

REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)

- + Programación (Curso 1 / Cuatrimestre 1)
- + Estructura de Computadores (Curso 2 / Cuatrimestre 1)
- + Sistemas Operativos (Curso 2 / Cuatrimestre 2)

OBJETIVOS

El objetivo principal de esta asignatura es que el estudiante conozca los conceptos básicos sobre la arquitectura de un computador y el impacto que estos tienen sobre el rendimiento de las aplicaciones y los sistemas informáticos.

Para alcanzar este objetivo:

- + Se profundiza en aspectos avanzados de la arquitectura del computador como la optimizaciones de la jerarquía de memoria, el paralelismo a nivel de instrucción, o el diseño de multiprocesadores.
- + Se refuerzan conceptos básicos de programación concurrente y su interrelación con la arquitectura del computador.
- + Se introducen conceptos de programación paralela.
- + Se fomenta una preocupación constante por el rendimiento de las aplicaciones.

DESCRIPCIÓN DE CONTENIDOS: PROGRAMA

1. Fundamentos del diseño de computadores.
 - 1.1. Introducción.
 - 1.2. Perspectiva histórica.
 - 1.3. Clasificación de computadores.
 - 1.4. Paralelismo.
 - 1.5. Arquitectura del Computador.
2. Evaluación del rendimiento de sistemas informáticos.
 - 2.1. Clasificación de arquitecturas y evaluación.
 - 2.2. Fiabilidad de Sistemas Informáticos.
3. Jerarquía de memoria.
 - 3.1. Optimizaciones de memoria caché.
 - 3.2. Optimizaciones avanzadas de memoria caché.
4. Paralelismo a nivel de instrucción.
 - 4.1. Introducción al paralelismo a nivel de instrucción.
 - 4.2. Explotación del paralelismo a nivel de instrucción
5. Multiprocesadores.
 - 5.1. Arquitecturas de memoria compartida simétrica.
 - 5.2. Memoria compartida distribuida.
 - 5.3. Sincronización en memoria compartida.
 - 5.4. Modelos de consistencia de memoria.

- 6. Modelos de programación paralela y concurrente.
- 6.1. Introducción a la programación paralela: OpenMP.
- 6.2. Modelo de programación concurrente: Hilos en ISO C++.
- 6.3. Modelos de consistencia de memoria en C++.

ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

** ACTIVIDADES FORMATIVAS **

CLASES TEÓRICO-PRÁCTICAS [42 horas con 100% de presencialidad, 1.64 ECTS].

En ellas se presentarán los conocimientos que deben adquirir los alumnos. Estos recibirán las notas de clase y tendrán textos básicos de referencia para facilitar el seguimiento de las clases y el desarrollo del trabajo posterior. Se resolverán ejercicios por parte del alumno que le servirá de autoevaluación y para adquirir las capacidades necesarias. Clases de problemas, en las que se desarrollen y discutan los problemas que se proponen a los alumnos.

TALLERES Y/O PRÁCTICAS DE LABORATORIO [6 horas con 100% de presencialidad, 0.23 ECTS].

TUTORÍAS [28 horas con 25% de presencialidad, 1.09 ECTS].

Asistencia individualizada (tutorías individuales) o en grupo (tutorías colectivas) a los estudiantes por parte del profesor.

TRABAJO INDIVIDUAL O EN GRUPO DEL ESTUDIANTE [74 horas con 0% de presencialidad, 2.88 ECTS].

EXAMEN FINAL [4 horas 100% de presencialidad, 0.16 ECTS]

Se valorarán de forma global los conocimientos, destrezas y capacidades adquiridas a lo largo del curso.

** METODOLOGÍAS **

TRABAJO PREPARATIVO Y PREVIO. Análisis de ejercicios y lectura crítica de textos recomendados por el profesor de la asignatura: Ejercicios, artículos, informes, manuales, bien para su posterior discusión en clase, bien para ampliar y consolidar los conocimientos de la asignatura.

CLASE MAGISTRAL. Exposiciones en clase del profesor con soporte de medios informáticos y audiovisuales, en las que se desarrollan los conceptos principales de la materia y se proporcionan los materiales y la bibliografía para complementar el aprendizaje de los alumnos.

SESIONES PRÁCTICAS. Resolución de casos prácticos, problemas, etc. ¿ planteados por el profesor de manera individual o en grupo.

PRÁCTICAS DE LABORATORIO. Realización de diseños, montajes, sistemas, etc. en laboratorios bajo la orientación y supervisión del profesor

TRABAJOS E INFORMES. Elaboración de trabajos e informes de manera individual o en grupo

SISTEMA DE EVALUACIÓN

Peso porcentual del Examen Final: 40

Peso porcentual del resto de la evaluación: 60

Evaluaciones de conocimientos durante el curso: 24%

Prácticas realizadas individualmente o en grupo: 36%

Examen final: 40%

CONVOCATORIA ORDINARIA

Nota mínima en el examen: 3.5 puntos sobre 10.

Nota mínima en proyecto: 2 sobre 10.

La nota final se incrementará en 1 punto a aquellos alumnos que realicen todas las pruebas de evaluación continua, obtengan más un 7 en la evaluación continua y al menos 6 puntos en el examen final.

Para aquellos alumnos que no hayan completado el proceso de evaluación continua, o no hayan

Peso porcentual del Examen Final: 40

Peso porcentual del resto de la evaluación: 60

superado las notas mínimas de prácticas, el examen final tendrá un valor del 60% de la calificación total de la asignatura. Por tanto, para poder aprobar, el alumno deberá obtener una calificación superior a 8,33 sobre 10 en este examen.

BIBLIOGRAFÍA BÁSICA

- Hennessy, JL y Patterson, DA. Computer Architecture: A Quantitative Approach. 6th Edition., Morgan Kaufmann,, 2017

BIBLIOGRAFÍA COMPLEMENTARIA

- David A. Patterson, John Hennessy Computer Organization and Design MIPS Edition: The Hardware/Software Interface, Morgan Kaufmann, 2020

- Timothy G. Mattson, Yun (Helen) He and Alice E. Koniges The OpenMP Common Core: Making OpenMP Simple Again, MIT Press, 2019

- Williams, A. C++ Concurrency in Action. Practical Multithreading. 2nd Edition, Manning., 2018