

Arquitectura de Computadores y Sistemas Operativos

CRÉDITOS: 3 ECTS

PROFESOR/A COORDINADOR/A: Juan Carlos Pichel Campos (juancarlos.pichel@usc.es)

UNIVERSIDAD DESDE LA QUE IMPARTE EL PROFESOR/A COORDINADOR/A: USC

¿HA DADO O VA A DAR AUTORIZACIÓN PARA GRABAR LAS CLASES DE ESTA ASIGNATURA? Si

CONTENIDOS:

1. Introducción y evolución de la tecnología de los computadores.
2. Representación y procesamiento de la información.
3. Arquitectura de microprocesadores.
4. Jerarquía de memoria.
5. Introducción a los Sistemas operativos.

METODOLOGÍA:

1. Planificación de los contenidos de cada clase.
2. Explicación en el encerado (lección magistral).
3. Entrega de boletines de problemas.
4. Programación en ensamblador.
5. Programación con llamadas a funciones del sistema operativo.

IDIOMA: El idioma se adaptará en función del auditorio.

¿SE REQUIERE PRESENCIALIDAD PARA ASISTIR A LAS CLASES? Videoconferencia.

BIBLIOGRAFÍA:

Patterson e Hennessy. Estructura y Diseño de Computadores: Interficie circuitería/programación. Ed. Reverté 2000.

Hennessy e Patterson. Computer architecture, a quantitative approach. 4th edition. Morgan Kaufmann Pub. 2006. ISBN 978-0-12-370490-0.

A. S. Tanenbaum. Sistemas Operativos Modernos (2a edición). Editorial Prentice-Hall, 2003. ISBN 970-26-0315-3.

COMPETENCIAS

Básicas y generales:

CG1: Poseer conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación, sabiendo traducir necesidades industriales en términos de proyectos de I+D+i en el campo de la Matemática Industrial.

CG3: Ser capaz de integrar conocimientos para enfrentarse a la formulación de juicios a partir de información que, aún siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos.

Específicas:

CE4: Ser capaz de seleccionar un conjunto de técnicas numéricas, lenguajes y herramientas informáticas, adecuadas para resolver un modelo matemático.

CE5: Ser capaz de validar e interpretar los resultados obtenidos, comparando con visualizaciones, medidas experimentales y/o requisitos funcionales del correspondiente sistema físico/de ingeniería.

De especialización "Simulación Numérica":

CS2: Saber adaptar, modificar e implementar herramientas de software de simulación numérica.

¿SE VA A USAR ALGÚN TIPO DE PLATAFORMA VIRTUAL? No.

¿SE NECESITA ALGÚN SOFTWARE ESPECÍFICO? No.

CRITERIOS PARA LA 1ª OPORTUNIDAD DE EVALUACIÓN:

Se propondrán prácticas y problemas que serán presentadas y evaluadas contribuyendo en un 30% de la nota final.

Para poder aprobar la asignatura la nota de estos ejercicios y problemas debe ser igual o superior a 5.

Se realizará un examen que supondrá el restante 70% de la calificación final.

CRITERIOS PARA LA 2ª OPORTUNIDAD DE EVALUACIÓN:

Para poder aprobar la asignatura en la 2ª oportunidad los alumnos deberán haber obtenido una nota superior o igual a 5 en la parte de prácticas y problemas. Esta nota supondrá el 30% de la nota final.

Se realizará un examen que supondrá el restante 70% de la calificación final.

Cálculo Paralelo

CRÉDITOS: 3 ECTS

PROFESOR/A COORDINADOR/A: José A. Alvarez Dios (joseantonio.alvarez.dios@usc.es)

UNIVERSIDAD DESDE LA QUE IMPARTE EL PROFESOR/A COORDINADOR/A: USC

¿HA DADO O VA A DAR AUTORIZACIÓN PARA GRABAR LAS CLASES DE ESTA ASIGNATURA? No

PROFESOR 1: Carlos Fernández Sánchez (carlosf@cesga.es)

UNIVERSIDAD DESDE LA QUE IMPARTE EL PROFESOR/A: USC

¿HA DADO O VA A DAR AUTORIZACIÓN PARA GRABAR LAS CLASES DE ESTA ASIGNATURA? No

CONTENIDOS:

Programación de ordenadores paralelos usando MPI y OpenMP bajo C y Fortran.

METODOLOGÍA:

Una vez entendida la exposición teórica, el alumno debe comprender los ejemplos de programas expuestos en clase para elaborar a continuación sus propios programas.

IDIOMA: Castellano

¿SE REQUIERE PRESENCIALIDAD PARA ASISTIR A LAS CLASES? Videoconferencia, Desde la universidad que emite el profesor.

BIBLIOGRAFÍA:

Parallel Programming in C with MPI and OpenMP. Michael J. Quinn (McGraw-Hill Science/Engineering/Math, 2003).

Introduction to Parallel Computing, Second Edition, by Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar (Addison -Wesley, 2003).

Parallel Programming with MPI, by Peter Pacheco (Morgan Kauffman Publishers, 1997).

Parallel Programming, by Barry Wilkinson and Michael Allen (Prentice Hall, 1999).

COMPETENCIAS

Básicas y generales:

CG1: Poseer conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación, sabiendo traducir necesidades industriales en términos de proyectos de I+D+i en el campo de la Matemática Industrial.

CG3: Ser capaz de integrar conocimientos para enfrentarse a la formulación de juicios a partir de información que, aún siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos.

Específicas:

CE4: Ser capaz de seleccionar un conjunto de técnicas numéricas, lenguajes y herramientas informáticas, adecuadas para resolver un modelo matemático.

CE5: Ser capaz de validar e interpretar los resultados obtenidos, comparando con visualizaciones, medidas experimentales y/o requisitos funcionales del correspondiente sistema físico/de ingeniería.

De especialización "Simulación Numérica":

CS2: Saber adaptar, modificar e implementar herramientas de software de simulación numérica.

¿SE VA A USAR ALGÚN TIPO DE PLATAFORMA VIRTUAL? No.

¿SE NECESITA ALGÚN SOFTWARE ESPECÍFICO? Si. MPI, OpenMP, compiladores C y Fortran.

CRITERIOS PARA LA 1ª OPORTUNIDAD DE EVALUACIÓN:

Trabajos y examen.

CRITERIOS PARA LA 2ª OPORTUNIDAD DE EVALUACIÓN:

Trabajos y examen.

Programación en C++

CRÉDITOS: 3 ECTS

PROFESOR/A COORDINADOR/A: Ana María Ferreiro Ferreiro (ana.ferreiro@udc.es)

UNIVERSIDAD DESDE LA QUE IMPARTE EL PROFESOR/A COORDINADOR/A: UDC

¿HA DADO O VA A DAR AUTORIZACIÓN PARA GRABAR LAS CLASES DE ESTA ASIGNATURA? Si

PROFESOR 1: José Antonio García Rodríguez (jose.garcia.rodriguez@udc.es)

UNIVERSIDAD DESDE LA QUE IMPARTE EL PROFESOR/A: UDC

¿HA DADO O VA A DAR AUTORIZACIÓN PARA GRABAR LAS CLASES DE ESTA ASIGNATURA? Si

CONTENIDOS:

Tema 1: El lenguaje de programación C++.

- Introducción a la programación en C++.
- Tipos de datos básicos.
- I/O por teclado y por fichero.
 - Sentencias de control.
 - Gestión dinámica de memoria: punteros.
 - Estructuras.
 - Funciones. Sobrecarga.

Tema 2: Programación Orientada a Objetos en C++.

- Introducción a la Programación Orientada a Objetos.
- Clases e instancias.
- Sobrecarga de operadores.

- Funciones y clases friend.
- Herencia.
- Polimorfismo.
- Templates (plantillas).

Tema 3: Standard Template Library (STL).

- Introducción a la STL.
- Contenedores e iteradores.
- Manejo de contenedores básicos.

METODOLOGÍA:

a) Sesión magistral: En las clases teóricas se explicará la sintaxis del lenguaje de programación C++, se abordará la Programación Orientada a Objetos, así como la sintaxis para expresar los conceptos de la POO en C++.

b) Prácticas de laboratorio: Clases de prácticas tuteladas en las que los alumnos podrán en práctica mediante pequeños ejercicios los conceptos vistos en las clases teóricas.

Se intercalarán las explicaciones teóricas con las prácticas, con el objetivo de facilitar el aprendizaje.

c) Trabajos tutelados: Proyectos y ejercicios a realizar individualmente por el alumno para profundizar en la comprensión de la materia aplicados al cálculo numérico.

IDIOMA: Castellano

¿SE REQUIERE PRESENCIALIDAD PARA ASISTIR A LAS CLASES? Videoconferencia, No se requiere presencialidad.

BIBLIOGRAFÍA:

- Harvey M. Deitel, Paul J. Deitel (2009). C++ : cómo programar (6ª ed.). Pearson Educación
 - Bjarne Stroustrup ([2001] [2007 reimp.]). El Lenguaje de programación C++ . Addison-Wesley Iberoamericana
 - Walter Savitch (2004). Problem Solving with C++: The Object of - Programming, Fifth Edition . Addison-Wesley
 - Ray Lischner (2003). C++ In a Nutshell. O'Reilly Media.
-

COMPETENCIAS

Básicas y generales:

CG1: Poseer conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación, sabiendo traducir necesidades industriales en términos de proyectos de I+D+i en el campo de la Matemática Industrial.

CG3: Ser capaz de integrar conocimientos para enfrentarse a la formulación de juicios a partir de información que, aún siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos.

Específicas:

CE4: Ser capaz de seleccionar un conjunto de técnicas numéricas, lenguajes y herramientas informáticas, adecuadas para resolver un modelo matemático.

CE5: Ser capaz de validar e interpretar los resultados obtenidos, comparando con visualizaciones, medidas experimentales y/o requisitos funcionales del correspondiente sistema físico/de ingeniería.

De especialización "Simulación Numérica":

CS2: Saber adaptar, modificar e implementar herramientas de software de simulación numérica.

¿SE VA A USAR ALGÚN TIPO DE PLATAFORMA VIRTUAL? Si. Google groups / Google sites

¿SE NECESITA ALGÚN SOFTWARE ESPECÍFICO? Si. Linux. Compiladores gnu: gcc, g++, gnumake. Un IDE de desarrollo para C++, preferentemente KDEVELOP o alternativamente NETBEANS para C++.

CRITERIOS PARA LA 1ª OPORTUNIDAD DE EVALUACIÓN:

La evaluación se realizará sólo mediante diferentes trabajos prácticos, todos ellos de entrega obligatoria.

Se propondrán trabajos semanalmente.

Se propondrá una práctica final, donde se apliquen todos los conceptos de POO estudiados en la asignatura.

CRITERIOS PARA LA 2ª OPORTUNIDAD DE EVALUACIÓN:

La evaluación se realizará sólo mediante diferentes trabajos prácticos, todos ellos de entrega obligatoria.

Se propondrán trabajos semanalmente.

Se propondrá una práctica final, donde se apliquen todos los conceptos de POO estudiados en la asignatura.

COMENTARIOS:

Es recomendable que el alumno tenga conocimientos básicos de programación

Redes de Computadores y Computación Distribuida

CRÉDITOS: 3 ECTS

PROFESOR/A COORDINADOR/A: Jesús María Rodríguez Presedo (jesus.presedo@usc.es)

UNIVERSIDAD DESDE LA QUE IMPARTE EL PROFESOR/A COORDINADOR/A: USC

¿HA DADO O VA A DAR AUTORIZACIÓN PARA GRABAR LAS CLASES DE ESTA ASIGNATURA? Si

PROFESOR 1: José Carlos Cabaleiro Domínguez (jc.cabaleiro@usc.es)

UNIVERSIDAD DESDE LA QUE IMPARTE EL PROFESOR/A: USC

¿HA DADO O VA A DAR AUTORIZACIÓN PARA GRABAR LAS CLASES DE ESTA ASIGNATURA? Si

CONTENIDOS:

1. Redes [5 horas].

- * El protocolo TCP/IP. Modelo de capas.
- * Servicios orientados a conexión y sin conexión. TCP y UDP.
- * Fundamentos de la transmisión fiable.
- * Clasificación de redes.
- * Redes de acceso residencial. Módem, ADSL, cable.
- * Redes de area local. Ethernet.
- * Hubs, switches y bridges.

2. Paradigmas de la computación distribuida [5 horas].

- * El paradigma cliente-servidor.
- * Las arquitecturas P2P.
- * El modelo publica-suscribe.

- * Invocación remota.
- * El paradigma de objetos distribuidos.
- * Otros paradigmas de computación distribuida.

Prácticas:

1. Direcciones IP, puertos, protocolos, DNS (2 horas).
2. Sockets orientados a conexión, TCP (4 horas).
3. Sockets sin conexión, UDP (4 horas).
4. Remote Method Invocation (RMI) (6 horas).
5. Introducción a la programación de aplicaciones web (4 horas).

METODOLOGÍA:

Clases de teoría magistrales y clases prácticas en el aula de informática.

IDIOMA: Castellano.

¿SE REQUIERE PRESENCIALIDAD PARA ASISTIR A LAS CLASES? Videoconferencia.

BIBLIOGRAFÍA:

1. J.F. Kurose y K.W. Ross, "Redes de Computadores. Un enfoque descendente", 5ª edición ISBN: 9848478291199, 2010, Pearson Educación S. A.
 2. D.E. Comer, D.L. Stevens y M. Evangelista, "Internetworking with TCP/IP, Vol. III: Client-Server Programming and Applications, Linux/Posix Sockets Version", ISBN: 0130320714, 2001, Prentice Hall.
 3. M. L. Liu. "Computación Distribuida: Fundamentos y aplicaciones". Addison Wesley 2004. ISBN 84-7829-066-4.
 4. G. Coulouris, J. Dollimore y T. Kindberg. "Sistemas Distribuidos: conceptos y diseño". Addison Wesley 2001. ISBN 84-7829-049-4.
 5. M. Hall y L. Brown. "Core Web programming. Segunda edición". Prentice Hall 2001. ISBN 0-13-089793-0.
-

COMPETENCIAS

Básicas y generales:

CG1: Poseer conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación, sabiendo traducir necesidades industriales en términos de proyectos de I+D+i en el campo de la Matemática Industrial.

CG3: Ser capaz de integrar conocimientos para enfrentarse a la formulación de juicios a partir de información que, aún siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos.

Específicas:

CE4: Ser capaz de seleccionar un conjunto de técnicas numéricas, lenguajes y herramientas informáticas, adecuadas para resolver un modelo matemático.

CE5: Ser capaz de validar e interpretar los resultados obtenidos, comparando con visualizaciones, medidas experimentales y/o requisitos funcionales del correspondiente sistema físico/de ingeniería.

De especialización "Simulación Numérica":

CS2: Saber adaptar, modificar e implementar herramientas de software de simulación numérica.

¿SE VA A USAR ALGÚN TIPO DE PLATAFORMA VIRTUAL? Si. Campus Virtual USC (Moodle).

¿SE NECESITA ALGÚN SOFTWARE ESPECÍFICO? Si. Entorno de desarrollo para Java (Eclipse o Netbeans).

CRITERIOS PARA LA 1ª OPORTUNIDAD DE EVALUACIÓN:

Superación de un examen que representará el 50% de la nota final. Será obligatorio obtener un 5 sobre 10 para aprobar.

La asistencia a las prácticas y participación en las clases de debate supondrá el 20% de la nota final.

La calidad de los trabajos así como su adecuada exposición supondrá el 30% de la nota final.

CRITERIOS PARA LA 2ª OPORTUNIDAD DE EVALUACIÓN:

Mismos criterios que en la 1ª oportunidad de evaluación.
