

Plan 512 GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE TELECOMUNICACIÓN

Asignatura 46673 SISTEMAS DE TIEMPO REAL

Tipo de asignatura (básica, obligatoria u optativa)

Optativa

Créditos ECTS

6 ECTS

Competencias que contribuye a desarrollar

GENERALES

- GBE1. Capacidad para manejar especificaciones, reglamentos y normas de obligado cumplimiento.
- GBE3. Capacidad para resolver problemas con iniciativa, creatividad y razonamiento crítico.
- GBE4. Capacidad para diseñar y llevar a cabo experimentos, así como analizar e interpretar datos.
- GBE5. Capacidad para elaborar informes basados en el análisis crítico de la bibliografía técnica y de la realidad en el campo de su especialidad.
- GE2. Capacidad para trabajar en un grupo multidisciplinar y multilingüe, responsabilizándose de la dirección de actividades objeto de los proyectos del ámbito de su especialidad y consiguiendo resultados eficaces.
- GE3. Capacidad para desarrollar metodologías y destrezas de aprendizaje autónomo eficiente para la adaptación y actualización de nuevos conocimientos y avances científicos.
- GC1. Capacidad de organización, planificación y gestión del tiempo.
- GC2. Capacidad para comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.
- GC3. Trabajar en cualquier contexto, individual o en grupo, de aprendizaje o profesional, local o internacional, desde el respeto a los derechos fundamentales, de igualdad de sexo, raza o religión y los principios de accesibilidad universal, así como la cultura de paz.

ESPECÍFICAS

- T2. Capacidad de utilizar aplicaciones informáticas (ofimáticas, bases de datos, cálculo avanzado, gestión de proyectos, visualización, etc.) para apoyar el desarrollo y explotación de redes, servicios y aplicaciones de telecomunicación y electrónica.
- T3. Capacidad para utilizar herramientas informáticas de búsqueda de recursos bibliográficos o de información relacionada con las telecomunicaciones y la electrónica.
- T12. Conocimiento y utilización de los conceptos de arquitectura de red, protocolos e interfaces de comunicaciones.
- T13. Capacidad de diferenciar los conceptos de redes de acceso y transporte, redes de conmutación de circuitos y de paquetes, redes fijas y móviles, así como los sistemas y aplicaciones de red distribuidos, servicios de voz, datos, audio, video y servicios interactivos y multimedia.
- TEL3. Capacidad de construir, explotar y gestionar servicios telemáticos, utilizando herramientas analíticas de planificación, de dimensionado y de análisis.
- TEL5. Capacidad de seguir el progreso tecnológico de transmisión, conmutación y proceso para mejorar las redes y servicios telemáticos.
- TEL7. Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas.

Objetivos/Resultados de aprendizaje

Al finalizar la asignatura el alumno deberá ser capaz de:

- -Comprender la naturaleza específica de los sistemas de tiempo real.
- Conocer los requisitos de los lenguajes de programación para el diseño de sistemas de tiempo real.
- Adaptar los principales mecanismos que poseen los lenguajes de programación para la implementación de servicios y aplicaciones de tiempo real.

- Identificar, implementar y valorar la importancia de la concurrencia de procesos en sistemas de control software y afrontar la problemática derivada.
- Conocer los modelos básicos de programación con procesos e hilos y su aplicación a sistemas concretos de tiempo real.
- Desarrollar sistemas de tiempo real no estrictos usando ordenadores embebidos tipo PC.
- Entender las diferencias entre paralelismo a nivel de tarea y paralelismo a nivel de datos.
- Conocer las bases de la programación de procesadores masivamente paralelos.

Contenidos

Bloque 1:
Introducción general

Carga de trabajo en créditos ECTS:
0.4

a.
Contextualización y justificación

Este bloque temático introduce y relaciona los contenidos que se estudiarán en los siguientes bloques temáticos, a fin de que el alumno adquiera una visión global de la asignatura ya en las primeras semanas. Con esta intención, se tratarán superficialmente los conceptos y aspectos tecnológicos de toda la asignatura, facilitando al alumno la tarea de asociar entre sí los distintos contenidos, como paso previo al estudio detallado de cada uno de ellos.

b.
Objetivos de aprendizaje

Al finalizar este bloque temático, el alumno deberá ser capaz de:

- Comprender la naturaleza específica de los sistemas de tiempo real.
- Conocer los requisitos de los lenguajes de programación para el diseño de sistemas de tiempo real.
- Demostrar capacidades de búsqueda de información en cuanto a los componentes de un sistema de tiempo real.

c.
Contenidos

TEMA 0: Presentación de la asignatura

0.1 Descripción de competencias y contenidos

0.2 Objetivos de aprendizaje

0.3 Herramientas disponibles

0.4 Sistema de evaluación

TEMA 1: Introducción a los sistemas de tiempo real (STR)

1.1 Conceptos y aspectos generales

1.2 Sistemas operativos y lenguajes de programación para STR

1.3 Diseño de sistemas de tiempo real

1.4 Ejemplos de STR

d.
Métodos docentes

- Clase magistral participativa

e.
Plan de trabajo
Véase el Anexo I.

f.
Evaluación

La evaluación de la adquisición de competencias se basará en:

- Valoración de la actitud y participación del alumno en las actividades formativas.
- Capacidad para identificar los componentes fundamentales de un sistema de tiempo real.

g.
Bibliografía básica

- Burns, A., Wellings, A., Real-Time Systems and Programming Languages (4th Edition), Addison Wesley, 2009. ISBN: 978-0321417459.
- Jiménez García, L.M., Puerto Manchón, R., Reinoso García, O., Sistemas informáticos de tiempo real. Universidad Miguel Hernández, 2000. ISBN: 84-95315-51-3.

h.
Bibliografía complementaria

- Welling, A., Concurrent and Real-Time Programming in Java, John Wiley & Sons, 2004. ISBN: 978-0470844373.
- Burns, A. y Wellings, A., Concurrent and Real-Time Programming in Ada (3rd Edition), Cambridge University Press, 2007. ISBN: 978-0521866972.
- Buhr, R.J.A.. An Introduction to Real-Time Systems, Prentice Hall, 1998. ISBN: 978-0136060703.
- Tanenbaum, A. S., Sistemas operativos modernos, Prentice-Hall, 2009. ISBN: 978-6074420463.
- Stalling, W. Sistemas Operativos: Principios de diseño e interioridades (5ª Edición), Prentice Hall, 2006. ISBN: 978-8420544625.

i.
Recursos necesarios

Serán necesarios los siguientes recursos, todos ellos facilitados por la UVa o el profesor:

- Entorno de trabajo en la plataforma Moodle.
- Documentación de apoyo.
- Laboratorio con sistema operativo Linux.

Bloque 2:
Lenguajes para sistemas embebidos tipo PC

Carga de trabajo en créditos ECTS:
0.8

a.
Contextualización y justificación

En este bloque temático se realiza un repaso de aspectos avanzados de programación en lenguaje C, que servirá para adentrarse en el siguiente bloque en la programación con llamadas al sistema tipo POSIX en Unix/Linux. Adicionalmente, se realizará una introducción al lenguaje de programación interpretado Python y una introducción al funcionamiento del sistema embebido tipo PC Raspberry Pi, para el que existen multitud de librerías disponibles en este lenguaje.

b.
Objetivos de aprendizaje

Al finalizar este bloque temático, el alumno deberá ser capaz de:

- Realizar programas con características avanzadas del lenguaje C.
- Realizar programas básicos en lenguaje Python.
- Instalar el sistema operativo en un ordenador de bajo coste Raspberry Pi.

c.
Contenidos

- TEMA 2: Programación avanzada en lenguaje C
 - 2.1 Programación incremental utilizando make
 - 2.2 Punteros multidimensionales y reserva dinámica de memoria
 - 2.3 Punteros a funciones
 - 2.4 Ejercicios de programación
- TEMA 3: Introducción al lenguaje Python
 - 3.1 Introducción al lenguaje y tipos de datos
 - 3.2 Funciones y programación orientada a objetos
 - 3.3 Manejo de E/S y excepciones
 - 3.4 Módulos y paquetes en Python
 - 3.5 Ejercicios de programación
- TEMA 4: Introducción a Raspberry Pi

4.1 Instalación y preparación de Raspberry Pi

4.2 Introducción al GPIO de Raspberry Pi

d.

Métodos docentes

- Clase magistral participativa
- Prácticas en laboratorio

e.

Plan de trabajo

Véase el Anexo I.

f.

Evaluación

La evaluación de la adquisición de competencias se basará en:

- Capacidad para desarrollar programas en C con funciones POSIX.
- Capacidad para desarrollar programas básicos en Python.
- Capacidad de planificación para la realización de un proyecto en grupo.
- Capacidad para conocer las ventajas e inconvenientes de un sistema embebido tipo PC frente a otras alternativas para Sistemas de Tiempo Real.

g.

Bibliografía básica

- Kernighan, B.W., Pike, R., El entorno de programación Unix, Prentice Hall, 1995. ISBN: 978-9688800676.
- Marzal, A. y Gracia, I. Introducción a la programación con Python, Universitat Jaume I. ISBN: 978-84-692-5869-9. Disponible en: <http://repositori.uji.es/xmlui/bitstream/handle/10234/102653/s93.pdf>

h.

Bibliografía complementaria

- Cox, T. Raspberry Pi Cookbook for Python Programmers, Packt Publishing, 2014. ISBN 978-1-84969-662-3.

i.

Recursos necesarios

Serán necesarios los siguientes recursos, todos ellos facilitados por la UVa o el profesor:

- Entorno de trabajo en la plataforma Moodle
- Documentación de apoyo.
- Laboratorio con sistema operativo Linux
- Ordenador Raspberry Pi

Bloque 3:

Concurrencia, comunicación y sincronización en sistemas Unix/Linux

Carga de trabajo en créditos ECTS:

2.0

a.

Contextualización y justificación

En este bloque temático se estudian los aspectos teóricos necesarios sobre sistemas concurrentes y los mecanismos asociados. Se estudiará la gestión y planificación de procesos e hilos del sistema operativo y los métodos disponibles para crear, comunicar y sincronizar procesos e hilos mediante el desarrollo de aplicaciones. Con tal fin, se introducen en este tema las llamadas al sistema con el estándar POSIX de programación en C y las librerías necesarias de Python para multiprocesamiento.

b.

Objetivos de aprendizaje

Al finalizar este bloque temático, el alumno deberá ser capaz de:

- Adaptar los principales mecanismos que poseen los lenguajes de programación para la implementación de servicios y aplicaciones de tiempo real.
-

- Identificar, implementar y valorar la importancia de la concurrencia de procesos en sistemas de control software y afrontar la problemática derivada.
- Conocer los modelos básicos de programación con procesos e hilos y su aplicación a sistemas concretos de tiempo real.
- Entender la necesidad de ciertas estrategias para la mejora del rendimiento de programas.

c.

Contenidos

TEMA 5: Procesos

- 5.1 Conceptos y aspectos generales
- 5.2 Creación de procesos
- 5.3 Comunicación y sincronización de procesos
- 5.4 Programación de procesos usando C (POSIX) y Python
- 5.5 Ejercicios de programación con GTI-clip (Raspberry Pi)

TEMA 6: Hilos

- 6.1 Conceptos y aspectos generales
- 6.2 Creación de hilos
- 6.3 Comunicación y sincronización de hilos
- 6.4 Programación de hilos usando C (POSIX) y Python
- 6.5 Ejercicios de programación con GTI-clip (Raspberry Pi)

d.

Métodos docentes

- Clase magistral participativa
- Prácticas en laboratorio

e.

Plan de trabajo

Véase el Anexo I.

f.

Evaluación

La evaluación de la adquisición de competencias se basará en:

- Capacidad para desarrollar programas en C con funciones POSIX.
- Capacidad para desarrollar programas en Python con módulos para concurrencia.
- Capacidad para evaluar el compromiso coste/rendimiento en la elección óptima de un sistema informático.

g.

Bibliografía básica

- Jiménez García, L.M., Puerto Manchón, R., Reinoso García, O., Sistemas informáticos de tiempo real. Universidad Miguel Hernández, 2000. ISBN: 84-95315-51-3.

h.

Bibliografía complementaria

- Mitchell, M., Oldham, J. and Samuel, A, Advanced Linux Programming, New Riders Publishing, 2001. ISBN 0-7357-1043-0. Disponible en: <http://advancedlinuxprogramming.com>
- Kernighan, B.W., Pike, R., El entorno de programación Unix, Prentice Hall, 1995. ISBN: 978-9688800676.
- Stallings, W. Sistemas Operativos: Principios de diseño e interioridades (5ª Edición), Prentice Hall, 2006. ISBN: 978-8420544625.
- Palach, J. Parallel Programming with Python, Packt Publishing Ltd, 2014. ISBN: 978-1-78328-839-7.
- Dimitriadis Y. y Díaz Pernas F. J. Introducción práctica a la administración de sistemas en Internet. Servicio de Publicaciones de la Universidad de Valladolid, 1998.

i.

Recursos necesarios

Serán necesarios los siguientes recursos, todos ellos facilitados por la UVa o el profesor:

- Entorno de trabajo en la plataforma Moodle
- Documentación de apoyo.
- Laboratorio con sistemas operativos Windows y Linux.
- Ordenador Raspberry Pi y placa de aprendizaje GTI-clip.

Bloque 4:

Diseño e implementación de un sistema de tiempo real no estricto con sensores y actuadores

Carga de trabajo en créditos ECTS:

1.2

a.

Contextualización y justificación

En este bloque temático se estudian los aspectos teóricos necesarios sobre sistemas concurrentes y los mecanismos asociados. Se estudiará la gestión y planificación de procesos e hilos del sistema operativo y los métodos disponibles para crear, comunicar y sincronizar procesos e hilos mediante el desarrollo de aplicaciones. Con tal fin, se introducen en este tema las llamadas al sistema con el estándar POSIX de programación en C y las librerías necesarias de Python para multiprocesamiento.

b.

Objetivos de aprendizaje

Al finalizar este bloque temático, el alumno deberá ser capaz de:

- Desarrollar sistemas de tiempo real no estrictos usando ordenadores embebidos tipo PC.

c.

Contenidos

TEMA 7: Sensores y actuadores para Raspberry Pi

7.1 Conceptos y aspectos generales

7.2 Ejemplos de programación con sensores y actuadores

7.3 Diseño de un sistema de tiempo real en grupo

d.

Métodos docentes

- Clase magistral participativa
- Prácticas en laboratorio
- Aprendizaje basado en proyectos
- Estudio del caso

e.

Plan de trabajo

Véase el Anexo I.

f.

Evaluación

La evaluación de la adquisición de competencias se basará en:

- Capacidad para identificar los componentes fundamentales de un sistema tiempo real.
- Capacidad para desarrollar programas en C con funciones POSIX y en Python con librerías de multiprocesamiento.
- Capacidad de planificación para la realización de un proyecto en grupo.

g.

Bibliografía básica

- Jiménez García, L.M., Puerto Manchón, R., Reinoso García, O., Sistemas informáticos de tiempo real. Universidad Miguel Hernández, 2000. ISBN: 84-95315-51-3.
- Cox, T. Raspberry Pi Cookbook for Python Programmers, Packt Publishing, 2014. ISBN 978-1-84969-662-3.

h.

Bibliografía complementaria

- Welling, A., Concurrent and Real-Time Programming in Java, John Wiley & Sons, 2004. ISBN: 978-0470844373.
- Burns, A. y Wellings, A., Concurrent and Real-Time Programming in Ada (3rd Edition), Cambridge University

i.
Recursos necesarios

Serán necesarios los siguientes recursos, todos ellos facilitados por la UVa o el profesor:

- Entorno de trabajo en la plataforma Moodle
- Documentación de apoyo.
- Laboratorio con sistemas operativos Windows y Linux.
- Ordenador Raspberry Pi, protoboard y componentes electrónicos.

Bloque 5:
Lenguajes para sistemas de procesamiento masivamente paralelos

Carga de trabajo en créditos ECTS:
1.6

a.
Contextualización y justificación

En este bloque temático se estudian los aspectos teóricos sobre procesadores masivamente paralelos tipo GPU (Graphics Processing Unit). Se proporcionará una visión histórica de la evolución de los procesadores gráficos hacia procesadores de cómputo general y de los principales campos de investigación y desarrollo en los que se están utilizando. Se introducirá al alumno en la programación de estos sistemas mediante los lenguajes de programación C y Python (CUDA).

b.
Objetivos de aprendizaje

Al finalizar este bloque temático, el alumno deberá ser capaz de:

- Entender las diferencias entre paralelismo a nivel de tarea y paralelismo a nivel de datos.
- Conocer las bases de la programación de procesadores masivamente paralelos.

c.
Contenidos

TEMA 8: Introducción al Procesamiento Paralelo en GPU
8.1 Historia y evolución
8.2 Diferencias con CPU
8.3 Desarrollo de aplicaciones
8.4 Ejercicios de programación sobre GPU

d.
Métodos docentes

- Clase magistral participativa
- Prácticas en laboratorio

e.
Plan de trabajo
Véase el Anexo I.

f.
Evaluación

La evaluación de la adquisición de competencias se basará en:

- Capacidad para identificar los componentes internos de una GPU.
- Capacidad para desarrollar programas en CUDA para C y Python con procesamiento en GPU.

g.
Bibliografía básica

· Sanders, J. and Kandrot, E., Cuda by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional, 2010. ISBN: 978-0131387683.

· Kirk, D.B., Mei, W., Programming Massively Parallel Processors, Second Edition: A Hands-on Approach, Morgan Kaufmann, 2012. ISBN: 978-0124159921.

h.

Bibliografía complementaria

· Welling, A., Concurrent and Real-Time Programming in Java, John Wiley & Sons, 2004. ISBN: 978-0470844373.

Principios Metodológicos/Métodos Docentes

Bloque 1:

- Clase magistral participativa

Bloque 2:

- Clase magistral participativa
- Prácticas en laboratorio

Bloque 3:

- Clase magistral participativa
- Prácticas en laboratorio

Bloque 4:

- Clase magistral participativa
- Prácticas en laboratorio
- Aprendizaje basado en proyectos
- Estudio del caso

Bloque 5:

- Clase magistral participativa
- Prácticas en laboratorio

Criterios y sistemas de evaluación

INSTRUMENTO/PROCEDIMIENTO

PESO EN LA NOTA FINAL

OBSERVACIONES

Pruebas objetivas sobre funcionamiento de los desarrollos realizados en los ejercicios de laboratorio de los bloques 2, 3 y 5.

40%

Con la entrega de cada enunciado se informará al alumno de cada uno de los aspectos susceptibles de ser evaluados y los criterios a emplear.

Pruebas objetivas sobre funcionamiento de los desarrollos realizados de forma grupal en el bloque 4.

20%

Con la entrega de cada enunciado se informará al alumno de cada uno de los aspectos susceptibles de ser evaluados y los criterios a emplear.

Entrega y exposición de trabajos sobre estudio del caso y sobre proyectos.

25%

Valoración de la presentación grupal realizada. Evaluación de la calidad del trabajo y las aptitudes transversales del alumno.

Actitud y participación.

15%

Valoración de la actitud en el laboratorio y participación activa en las clases y el campus virtual.

Recursos de aprendizaje y apoyo tutorial

Serán necesarios los siguientes recursos para cada bloque, todos ellos facilitados por la UVa o el profesor.

Bloque 1:

- Entorno de trabajo en la plataforma Moodle.
- Documentación de apoyo.
- Laboratorio con sistema operativo Linux.

Bloque 2:

- Entorno de trabajo en la plataforma Moodle.
- Documentación de apoyo.
- Laboratorio con sistema operativo Linux.
- Ordenador Raspberry Pi.

Bloque 3:

- Entorno de trabajo en la plataforma Moodle.
- Documentación de apoyo.
- Laboratorio con sistemas operativos Windows y Linux.
- Ordenador Raspberry Pi y placa de aprendizaje GTI-clip.

Bloque 4:

- Entorno de trabajo en la plataforma Moodle.
- Documentación de apoyo.
- Laboratorio con sistemas operativos Windows y Linux.
- Ordenador Raspberry Pi, protoboard y componentes electrónicos.

Calendario y horario

Segundo cuatrimestre:

- Lunes de 18 a 20 h.
- Jueves de 16 a 18 h.

Tabla de Dedicación del Estudiante a la Asignatura/Plan de Trabajo

ACTIVIDADES PRESENCIALES

HORAS

ACTIVIDADES NO PRESENCIALES

HORAS

Clases teórico-prácticas (T/M)

0

Estudio y trabajo autónomo individual

60

Clases prácticas de aula (A)

15

Estudio y trabajo autónomo grupal

30

Laboratorios (L)

30

Prácticas externas, clínicas o de campo

0

Seminarios (S)

10

Tutorías grupales (TG)

0

Evaluación (fuera del periodo oficial de exámenes)

5

Total presencial

60

Total no presencial

90

Responsable de la docencia (recomendable que se incluya información de contacto y breve CV en el que aparezcan sus líneas de investigación y alguna publicación relevante)

- MARIO MARTÍNEZ ZARZUELA.

E-mail: marmar@tel.uva.es

Despacho 2D006.

Extensión: 5702.

- DAVID GONZÁLEZ ORTEGA.

E-mail: davgon@tel.uva.es

Despacho 2D022.

Extensión: 5552.

Idioma en que se imparte

Español
