



## 1. Identificación

### 1.1. De la Asignatura

<b>Curso Académico</b>	2016/2017
<b>Titulación</b>	GRADO EN INGENIERIA INFORMÁTICA y PROG CONJUNTA DE ESTUDIOS OFICIALES GRADO MATEMÁTICAS Y GRADO ING. INFORMÁTICA
<b>Nombre de la Asignatura</b>	DISEÑO Y ESTRUCTURA INTERNA DE UN SISTEMA OPERATIVO
<b>Código</b>	3866
<b>Curso</b>	CUARTO y QUINTO(IC)
<b>Carácter</b>	OPTATIVA
<b>N.º Grupos</b>	2
<b>Créditos ECTS</b>	6
<b>Estimación del volumen de trabajo del alumno</b>	150
<b>Organización Temporal/Temporalidad</b>	1º Cuatrimestre y 1º Cuatrimestre(IC)
<b>Idiomas en que se imparte</b>	ESPAÑOL
<b>Tipo de Enseñanza</b>	Presencial

### 1.2. Del profesorado: Equipo Docente

<b>Coordinación de la asignatura</b>  JUAN PIERNAS CANOVAS  Grupo: 1 y 2	<b>Área/Departamento</b>	INGENIERÍA Y TECNOLOGÍA DE COMPUTADORES
	<b>Categoría</b>	PROFESORES TITULARES DE UNIVERSIDAD
	<b>Correo</b>	piernas@um.es
	<b>Electrónico / Página web / Tutoría electrónica</b>	http://ditec.um.es/personal/2  Tutoría Electrónica: Sí



	Teléfono, Horario y Lugar de atención al alumnado	Duración	Día	Horario	Lugar
		Primer Cuatrimestre	Martes	10:15- 12:15	868887657, Facultad de Informática B1.3.058
		Primer Cuatrimestre	Jueves	15:30- 16:30	868887657, Facultad de Informática B1.3.058
		Segundo Cuatrimestre	Martes	10:15- 13:15	868887657, Facultad de Informática B1.3.058

## 2. Presentación

Los objetivos formativos de esta asignatura son:

- Conocer las diversas posibilidades que existen a la hora de diseñar un sistema operativo.
- Comprender el funcionamiento de distintas técnicas avanzadas para la construcción eficiente de los diferentes subsistemas de un sistema operativo: procesos, memoria, ficheros, E/S y seguridad.
- Comprender el diseño y el funcionamiento de un sistema operativo microkernel.
- Conocer las peculiaridades de los sistemas operativos que se ejecutan en entorno multiprocesador.
- Ser capaces de aprovechar los conocimientos adquiridos para mejorar el rendimiento de un sistema operativo.
- Ser capaces de modificar, o añadir, componentes de un sistema operativo.
- Ser capaces de diseñar e implementar software de sistema.



Esta asignatura es una continuación natural de la asignatura *Ampliación de Sistemas Operativos* de 3º curso ya que desarrolla en profundidad muchos de los contenidos que se ven en ella. Además, introduce nuevos conceptos y mecanismos avanzados, especialmente, aquellos relacionados con el diseño y la implementación eficiente de un sistema operativo.

La asignatura permite al alumno comprender el funcionamiento de algunos de los mecanismos internos más importante de los sistemas operativos actuales, como la organización interna de los distintos bloques de un sistema operativo, la optimización de los sistemas de ficheros, la implementación de distintos mecanismos de planificación de procesos, la implementación del sistema de seguridad, siempre desde la perspectiva del diseñador de un sistema operativo. Este conocimiento le ayudará a analizar mejor el comportamiento de un sistema operativo, detectar posibles problemas de rendimiento y buscar soluciones, aspectos que se verán en la asignatura *Administración Avanzada de Sistemas Operativos*. Todo ello prepara al alumno para desempeñar adecuadamente distintas actividades profesionales, como administrador de sistemas, administrador de servidores de Internet, administrador de redes, etc.

Por último, la asignatura también persigue que el alumno desarrolle algunas capacidades transversales que le pueden ser útiles en su vida profesional, como la preparación y realización correctas de una exposición oral y el análisis crítico de documentos científicos.

### 3. Condiciones de acceso a la asignatura

#### 3.1 Incompatibilidades

#### 3.2 Recomendaciones

Es fundamental que el alumno haya cursado con éxito las asignaturas *Introducción a los Sistemas Operativos*, *Ampliación de Sistemas Operativos* y *Programación Concurrente y Distribuida*. Asimismo, es aconsejable que haya superado varias asignaturas de programación y de arquitectura de ordenadores en cursos anteriores.

### 4. Competencias

#### 4.1 Competencias Básicas

- CB2. Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.



- CB3. Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
- CB4. Que los estudiantes puedan transmitir información, ideas problemas y soluciones a un público tanto especializado como no especializado.

## 4.2 Competencias de la titulación

- CEII1: Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.
- CEII2: Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos.
- CEII3: Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan.
- CEII4: Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
- CEII6: Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.
- CEII7: Capacidad para conocer, comprender y aplicar la legislación necesaria durante el desarrollo de la profesión de Ingeniero Técnico en Informática y manejar especificaciones, reglamentos y normas de obligado cumplimiento.
- CEII8: Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
- CEII9: Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.
- CEII10: Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otrostrabajos análogos de informática.
- CEII11: Capacidad para analizar y valorar el impacto social y medioambiental de las soluciones técnicas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico en Informática.
- IC2: Capacidad de desarrollar procesadores específicos y sistemas empujados, así como desarrollar y optimizar el software de dichos sistemas.
- IC3: Capacidad de analizar y evaluar arquitecturas de computadores, incluyendo plataformas paralelas y distribuidas, así como desarrollar y optimizar software de para las mismas.
- IC4: Capacidad de diseñar e implementar software de sistema y de comunicaciones.
- IC5: Capacidad de analizar, evaluar y seleccionar las plataformas hardware y software más adecuadas para el soporte de aplicaciones empujadas y de tiempo real.

## 4.3 Competencias transversales y de materia

## 5. Contenidos

**TEMA 1. Diseño de Sistemas Operativos**

**TEMA 2. Arquitectura de sistemas operativos**

**TEMA 3. Gestión avanzada de procesos**



**TEMA 4. Gestión avanzada de memoria**

**TEMA 5. Sistemas de ficheros avanzados**

**TEMA 6. Gestión avanzada de E/S**

**TEMA 7. Seguridad y protección avanzada**

## PRÁCTICAS

### Práctica 1. Practicas de la asignatura: *Global*

Al iniciar el curso se propondrán a los alumnos una o más prácticas relacionadas con los conceptos vistos en teoría. El entorno de desarrollo será Linux. Las prácticas consistirán en realizar modificaciones a partes del núcleo de Linux o bien en añadir nuevas funcionalidades al mismo, bien directamente, bien a través de tecnologías como FUSE o similares.

## 6. Metodología Docente

Actividad Formativa	Metodología	Horas	Trabajo	Volumen
		Presenciales	Autónomo	de trabajo
Clase magistral		30	10	40
Trabajos de investigación		6	24	30
Prácticas		20	40	60
Examen de teoría		4	16	20
	Total	60	90	150

## 7. Horario de la asignatura

<http://www.um.es/web/informatica/contenido/estudios/grados/informatica/2016-17#horarios>

<http://www.um.es/web/matematicas/contenido/estudios/grados/pes/2016-17#horarios>



## 8. Sistema de Evaluación

<b>Métodos / Instrumentos</b>	Examen de teoría
<b>Criterios de Valoración</b>	La parte teórica se evalúa con un único examen tipo test que el alumno realizará de forma individual
<b>Ponderación</b>	0,25
<b>Métodos / Instrumentos</b>	Proyecto de prácticas
<b>Criterios de Valoración</b>	Se propondrán varias prácticas de las cuales el alumno tendrá que realizar una. Para la evaluación el alumno tendrá que entregar un proyecto de programación (memoria y código fuente) y pasar una entrevista personal, la cual determinará la nota final de la práctica. En la nota se tendrán en cuenta, entre otros aspectos, la documentación entregada y la corrección del código. Las prácticas se pueden realizar individualmente o en grupos de dos personas. En ningún caso se admitirán grupos de tres o más personas.
<b>Ponderación</b>	0,50



<b>Métodos / Instrumentos</b>	Trabajo de investigación
<b>Criterios de Valoración</b>	<p>El trabajo de investigación consiste en trabajar sobre un tema propuesto por el propio alumno o por el profesor. El alumno tendrá que hacer algunas de las siguientes actividades: documentación, análisis de posibles ventajas e inconvenientes, y evaluación si fuese posible.</p> <p>Para evaluar esta parte, el alumno deberá redactar un documento explicativo sobre el tema y hacer una presentación en clase de dicho tema. Para determinar la nota, se tendrá en cuenta la calidad tanto del documento como de la presentación.</p> <p>En el documento, los alumnos tendrán que reflejar los siguientes aspectos (la lista es orientativa ya que los detalles concretos dependerán del tema tratado):</p> <ul style="list-style-type: none"> <li>• Un resumen inicial del trabajo realizado.</li> <li>• Una introducción de aquellos conceptos que sean necesarios para entender la memoria.</li> <li>• Descripción y estudio comparativo de las distintas alternativas existentes.</li> <li>• Explicación de cómo poner en marcha el software analizado.</li> <li>• Ejemplos de prueba que pongan de manifiesto los puntos fuertes y débiles de cada alternativa analizada, etc.</li> </ul> <p>En las presentaciones, se valorará la organización de la exposición, la capacidad de mostrar los aspectos más interesantes, la duración de la exposición (si está ajustada o no al tiempo dado), etc.</p> <p>Los trabajos de investigación son exclusivos e individuales, es decir, dos o más alumnos no podrán realizar un mismo trabajo y el trabajo tendrá que ser realizado por una única persona.</p>
<b>Ponderación</b>	0,25

## Fechas de exámenes

<http://www.um.es/web/informatica/contenido/estudios/grados/informatica/2016-17#examenes>  
<http://www.um.es/web/matematicas/contenido/estudios/grados/pes/2016-17#examenes>

## 9. Bibliografía

### Bibliografía Básica



William Stallings. « Operating Systems: Internals and Design Principles », 7ª edición. Prentice Hall, 2012.



David A. Solomon, Mark E. Russinovich y Alex Ionescu. «Windows Internals». Microsoft Press, 5ª edición, 2012.



Robert Love. «Linux Kernel Development». 3ª edición, Addison-Wesley, 2010.

### Bibliografía Complementaria



Abraham Silberschatz, Peter B. Galvin, Greg Gagne. «Operating System Concepts». John Wiley & Sons, 8ª edición, 2009.



Andrew S. Tanenbaum y Albert S. Woodhull. «Operating Systems. Design and Implementation». Prentice Hall, 3ª edición, 2006.



Andrew S. Tanenbaum. «Sistemas Operativos Modernos». Prentice Hall, 3ª edición, 2009.



Gary Nutt. «Sistemas Operativos». Addison Wesley, 3ª edición, 2004.



Claudia Salzberg Rodriguez, Gordon Fischer y Steven Smolski. «The Linux Kernel Primer». Prentice Hall, 2005.

## 10. Observaciones y recomendaciones

La nota final de la asignatura se obtendrá de la media ponderada de las notas obtenidas en: la lectura y exposición de los artículos de investigación (25%), el proyecto de programación (50%) y el examen de teoría (25%). Cada parte se calificará con una nota de 0 a 10. No hay nota mínima en ninguna parte, es decir, la media ponderada se calculará incluso cuando la nota obtenida en una parte sea inferior a 5 o aunque el alumno no presente una parte (en este caso, a la hora de calcular la media, la nota de esta parte será 0).

La calificación en el acta será:

- Aprobado, notable, sobresaliente o matrícula de honor, lo que corresponda en cada caso, cuando la nota media sea igual o mayor que 5.
- No presentado, cuando la nota media sea inferior a 5 y el alumno haya obtenido 5 o más en cualquiera de las partes presentadas.



- Suspenso, cuando la nota media sea inferior a 5 y el alumno haya obtenido menos de 5 en alguna de las partes presentadas.

A continuación, se muestran algunos ejemplos de calificaciones:

- Un alumno aprobará la asignatura con un 5 si sólo presenta el proyecto de programación y obtiene un 10.
- Un alumno aprobará la asignatura con un 5,5 si obtiene un 9 en el proyecto de programación y en el artículo de investigación obtiene un 4.
- Un alumno aparecerá en el acta como "no presentado" si sólo entrega el proyecto de programación y obtiene un 6.
- Un alumno aparecerá en el acta como "suspenso" con un 3,75 si obtiene un 6 en el proyecto de programación y, además, presenta su trabajo sobre un artículo de investigación y obtiene un 3, ya que la media es inferior a 5 y no ha llegado al 5 en una de las partes entregadas.

La nota obtenida en cualquiera de las partes se guardará hasta la convocatoria de julio.