

GUÍA DOCENTE

2018/19

Centro 226 - Facultad de Informática

Ciclo Indiferente

Plan GINFOR20 - Grado en Ingeniería Informática

Curso 2º curso

ASIGNATURA

26022 - Introducción a los Sistemas Operativos

Créditos ECTS : 6

DESCRIPCIÓN Y CONTEXTUALIZACIÓN DE LA ASIGNATURA

La asignatura de Introducción a los Sistemas Operativos es una asignatura obligatoria incluida en el plan de estudios del grado de Ingeniería Informática y que se imparte en segundo curso. Otras asignaturas que completan la formación en sistemas operativos: Sistemas Operativos, Administración de Sistemas y Redes y Diseño de Sistemas Operativos y Tiempo Real se imparten dentro de la especialidad de Ingeniería de Computadores del Grado en Ingeniería Informática.

Cada una de las asignaturas anteriores plantea el aprendizaje de los sistemas operativos desde un punto de vista diferente y complementario. En particular, la asignatura Introducción a los Sistemas Operativos se centra en la descripción funcional de los sistemas operativos, a través de su interfaz de llamadas al sistema, que lo presenta al programador (de sistemas) como una máquina virtual que oculta en gran medida la complejidad del hardware subyacente. Por su parte, la asignatura Sistemas Operativos se centra en las políticas de gestión de los recursos, así como en la evaluación del rendimiento; la asignatura Administración de Sistemas y Redes se centra en aspectos de gestión de los sistemas operativos, y finalmente la asignatura Diseño de Sistemas Operativos y Tiempo Real se ocupa de la implementación del sistema.

Para el buen aprovechamiento de esta asignatura, es muy recomendable haber superado la asignatura de 'Estructura de Computadores' del 1º curso del grado.

COMPETENCIAS / RESULTADOS DE APRENDIZAJE DE LA ASIGNATURA

Al finalizar el curso, el estudiante deberá ser capaz de identificar las interfaces de un sistema operativo, y en particular de desarrollar sobre Linux utilidades para las aplicaciones a partir de su interfaz de llamadas al sistema. La asignatura hace énfasis en diferentes tipos de sistemas (tiempo compartido, tiempo real, etc), en sus conceptos fundamentales (ficheros, protección de accesos, procesos, elementos de comunicación) y en la funcionalidad de sus componentes (subsistemas para la gestión de procesos, memoria, entrada-salida).

Otros aspectos importantes que se van a potenciar son la capacidad de encontrar la información y las herramientas necesarias para resolver los problemas que se planteen, y la capacidad de describir con precisión la funcionalidad de las utilidades desarrolladas, de forma que puedan ser utilizadas por otros usuarios o desarrolladores.

Formalmente, de acuerdo al Plan de Estudios, además de las competencias de la asignatura, se trabajarán las competencias generales C4, C5, C8 y C9, así como las competencias RI5, RI10 y RI14 de la rama común informática, tal y como aparecen en el documento "Lista de competencias" en la dirección <http://www.ehu.es/documents/340468/516505/Lista+de+competencias.pdf>

CONTENIDOS TEORICO-PRACTICOS

Tema 1: Introducción.

Definición de sistema operativo como interfaz para las aplicaciones y como gestor de los recursos del computador. Perspectiva histórica de los sistemas operativos. Tipos de sistemas según su funcionalidad (monoprogramados, multiprogramados, de tiempo compartido, de tiempo real, empujados y distribuidos). Interfaces del sistema operativo (de usuario, de administración, de las aplicaciones, del desarrollador). Mercado actual de los sistemas operativos (sistemas propietarios, sistemas abiertos, software libre) y perspectiva. Prácticas: el Shell como interfaz de usuario y administración.

Tema 2: Mecanismo de llamadas al sistema.

Tomando como soporte los elementos que proporciona la interfaz hardware (espacios de direcciones, interfaz de los dispositivos, mecanismo de interrupciones), determinación de los servicios básicos comunes a establecer como código residente. Concepto de llamada al sistema operativo como función de acceso a un servicio residente. Implementación de las llamadas al sistema a través del mecanismo de interrupciones y los modos de ejecución. Prácticas: especificación en C de ejemplos de rutinas de acceso a servicios.

Tema 3: Entrada-salida y ficheros.

Concepto de independencia del dispositivo y redirección de la entrada-salida. Llamadas al sistema relacionadas con la entrada-salida. Modos de operación sobre dispositivos específicos y sobre ficheros. Concepto de sistema de ficheros y sus mecanismos de representación. Prácticas: uso de redirecciones desde el Shell, manejo del sistema de ficheros desde el Shell, programar ejemplos de entrada-salida.

Tema 4: Gestión de usuarios y seguridad.

Definición de sistema multiusuario. Tipos de usuarios, modos de acceso y dominios de protección. Mecanismos básicos de gestión de la protección de accesos en sistemas centralizados. Llamadas al sistema relacionadas con la protección de

accesos y la seguridad. Prácticas: manejo de derechos de acceso desde el Shell.

Tema 5: Gestión de la memoria

Carga y ubicación de programas en sistemas con uno o varios programas en memoria. Direccionamiento físico y virtual. Reubicación estática y dinámica. Código reentrante. Soporte para sistemas de memoria virtual. Librerías de enlace dinámico. Llamadas al sistema operativo relacionadas con la carga de programas y la gestión de la memoria. Prácticas: programación de un lanzador sencillo, monitorización de consumo de memoria (con programas reentrantes).

Tema 6: Control de procesos.

Concepto de flujo y contexto de ejecución. Sistemas multiprogramados y sistemas multihilo. Concepto de proceso (modelo Unix), estados y grafo de transición. Cambio de contexto. Planificación de procesos y políticas de planificación básicas. Llamadas al sistema operativo relacionadas con el control de procesos. Prácticas: lanzamiento de procesos background desde el Shell y monitorización de procesos, modificación del lanzador para hacerlo multiprogramado.

Tema 7: Comunicación y sincronización entre procesos

Conceptos de concurrencia, recurso compartido, condición de carrera y acceso exclusivo. Secciones críticas de código. Mecanismos básicos de acceso exclusivo a secciones críticas. Comunicación con paso de mensajes mediante buzones. Llamadas al sistema relacionadas con la comunicación entre procesos. Modelo de gestión de recursos basado en el esquema cliente-servidor. Ejemplos de gestores de recursos (drivers). Prácticas: comunicación de procesos (mediante pipes) desde el Shell, ejemplo sencillo de gestor.

METODOLOGÍA

La asignatura se desarrolla mediante un modelo de evaluación continua, basada en plantear una serie de problemas cuya resolución proporciona la adquisición de las competencias definidas. Como consecuencia, la división formal en horas de clase y de laboratorio no se seguirá de modo estricto.

TIPOS DE DOCENCIA

Tipo de Docencia	M	S	GA	GL	GO	GCL	TA	TI	GCA
Horas de Docencia Presencial	40			20					
Horas de Actividad No Presencial del Alumno	50			40					

Leyenda:

M: Macistral
GCL: P. Clínicas

S: Seminario
TA: Taller

GA: P. de Aula
TI: Taller Ind.

GL: P. Laboratorio
GCA: P. de Campo

GO: P. Ordenador

SISTEMAS DE EVALUACIÓN

- Sistema de evaluación continua
- Sistema de evaluación final

HERRAMIENTAS Y PORCENTAJES DE CALIFICACIÓN

- Prueba escrita a desarrollar 60%
- Realización de prácticas (ejercicios, casos o problemas) 40%

CONVOCATORIA ORDINARIA: ORIENTACIONES Y RENUNCIA

La preinscripción en el modo de evaluación continua se realizará en las fechas establecidas. La preinscripción pasará a ser definitiva tras la confirmación de la solicitud por parte del estudiante en las fechas que se establezcan y previa verificación del rendimiento parcial por parte del profesorado.

Para la parte de la asignatura basada en evaluación continua se basa en las siguientes modalidades de evaluación:

- Cuestionarios de autoevaluación y de evaluación individual (60%).
- Cuaderno con el código desarrollado, especificaciones, resultados de verificación y evaluación individual sobre el trabajo realizado (40%).

Para quienes no sigan la evaluación continua se prevé una prueba final que incluye preguntas de tipo teórico y realización de ejercicios prácticos (análisis de código, diseño/programación de utilidades...).

CONVOCATORIA EXTRAORDINARIA: ORIENTACIONES Y RENUNCIA

La convocatoria extraordinaria consistirá en una prueba final que incluye preguntas de tipo teórico y realización de ejercicios prácticos (análisis de código, diseño/programación de utilidades...).

MATERIALES DE USO OBLIGATORIO

Sistema operativo Linux, manuales y herramientas y programas C que se proporcionarán.

BIBLIOGRAFIA

Bibliografía básica

C. Rodríguez, I. Alegria, J. González, A. Lafuente: Descripción Funcional de los Sistemas Operativos. Síntesis, 1994.

F.M. Márquez: UNIX. Programación Avanzada 3ª Edición. Rama, 2004.

B.W. Kernighan, R. Pike: The Unix Programming Environment, Prentice-Hall, 1984.

M. Rochkind: Advanced Unix Programming, Addison-Wesley, 2004.

B.W. Kernighan, D.M.Ritchie, The C Programming Language, Prentice-Hall, 1988.

W. R Stevens, S. A. Rago, Advanced Programming in the UNIX® Environment Third Edition, 2013

Bibliografía de profundización

A. Silberschatz, P.B. Galvin, G. Gagne: Operating System Concepts (Eight edition), John Wiley & Sons, 2008.

W. Stallings: Sistemas Operativos (Quinta edición). Prentice-Hall, 2005.

A.S. Tanenbaum: Modern Operating Systems (3rd Edition), Prentice-Hall, 2008.

R. Arpaci-Dusseau, A. Arpaci-Dusseau: Operating Systems: Three Easy Pieces, Arpaci-Dusseau Books, 2015

Revistas

Direcciones de internet de interés

Sitio eGela de la asignatura

www.linux.org

www.gnu.org

OBSERVACIONES