



Guía docente 300028 - SO - Sistemas Operativos

Última modificación: 02/03/2016

Unidad responsable: Escuela de Ingeniería de Telecomunicación y Aeroespacial de Castelldefels
Unidad que imparte: 701 - DAC - Departamento de Arquitectura de Computadores.

Titulación: GRADO EN INGENIERÍA DE SISTEMAS DE TELECOMUNICACIÓN (Plan 2009). (Asignatura obligatoria).
GRADO EN INGENIERÍA TELEMÁTICA (Plan 2009). (Asignatura obligatoria).

Curso: 2014 **Créditos ECTS:** 6.0 **Idiomas:** Catalán, Castellano

PROFESORADO

Profesorado responsable: Definit a la infoweb de l'assignatura.

Otros: Definit a la infoweb de l'assignatura.

CAPACIDADES PREVIAS

La asignatura requiere conocimiento de programación en lenguaje C#.

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:

1. CE 27 TEL. Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas.(CIN/352/2009, BOE 20.2.2009)

Genéricas:

4. GESTIÓN DE PROYECTOS - Nivel 2: Definir los objetivos de un proyecto bien definido, de alcance reducido, y planificar su desarrollo, determinando los recursos necesarios, tareas a realizar, reparto de responsabilidades e integración. Utilizar adecuadamente herramientas de soporte a la gestión de proyectos.

7. USO EFICIENTE DE EQUIPOS E INSTRUMENTACIÓN - Nivel 1: Utilizar correctamente instrumental, equipos y software de los laboratorios de uso general o básicos. Realizar los experimentos y prácticas propuestos y analizar los resultados obtenidos.

Transversales:

2. APRENDIZAJE AUTÓNOMO - Nivel 2: Llevar a cabo las tareas encomendadas a partir de las orientaciones básicas dadas por el profesorado, decidiendo el tiempo que se necesita emplear para cada tarea, incluyendo aportaciones personales y ampliando las fuentes de información indicadas.

3. COMUNICACIÓN EFICAZ ORAL Y ESCRITA - Nivel 2: Utilizar estrategias para preparar y llevar a cabo las presentaciones orales y redactar textos y documentos con un contenido coherente, una estructura y un estilo adecuados y un buen nivel ortográfico y gramatical.

5. TERCERA LENGUA: Conocer una tercera lengua, que será preferentemente inglés, con un nivel adecuado de forma oral y por escrito y en consonancia con las necesidades que tendrán las tituladas y los titulados en cada enseñanza.

6. TRABAJO EN EQUIPO - Nivel 3: Dirigir y dinamizar grupos de trabajo, resolviendo posibles conflictos, valorando el trabajo hecho con las otras personas y evaluando la efectividad del equipo así como la presentación de los resultados generados.

8. USO SOLVENTE DE LOS RECURSOS DE INFORMACIÓN - Nivel 1: Identificar las propias necesidades de información y utilizar las colecciones, los espacios y los servicios disponibles para diseñar y ejecutar búsquedas simples adecuadas al ámbito temático.

METODOLOGÍAS DOCENTES

Es muy conveniente que los/las estudiantes tengan ordenador personal (idealmente portátil) con conexión a Internet.
No hay ningún requisito adicional.

El curso combina las siguientes metodologías docentes:

- Aprendizaje autónomo porque los estudiantes trabajarán el material de autoaprendizaje en casa.
- Aprendizaje cooperativo, porque los estudiantes se organizarán en pequeños grupos para realizar algunas de las tareas del curso.
- Aprendizaje basado en proyectos, porque los estudiantes desarrollarán un proyecto en equipo durante la segunda mitad del curso.
- Autoevaluación y evaluación entre compañeros de algunas de las entregas.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

Al acabar la asignatura de Sistemas Operativos, el/la estudiante debe ser capaz de:

- Utilizar los sistemas operativos de la familia de Linux a modo de usuario avanzado, tanto desde la línea de mandos como desde la interfaz de ventanas.
- Utilizar las llamadas a sistema de estándar POSIX desde programas escritos en C.
- Explicar el significado de los conceptos entrada/salida siguientes: dispositivo, canal, fichero, canales estándar, redirección, acceso secuencial, acceso directo, compartición.
- Explicar el significado de los conceptos de proceso, programa i thread (hilo de ejecución).
- Explicar el significado de los conceptos de socket y signal (interrupción software).
- Definir los permisos de acceso a un fichero y los dominios de protección.
- Saber resolver una aplicación de tipo cliente/servidor usando sockets, tanto con el protocolo TCP como UDP.
- Determinar, a partir de los conceptos de proceso y thread, qué modelo de programación paralela es más adecuado para resolver un problema.
- Crear y acceder a Bases de Datos Multinivel.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas aprendizaje autónomo	84,0	56.00
Horas grupo pequeño	43,0	28.67
Horas actividades dirigidas	23,0	15.33

Dedicación total: 150 h



CONTENIDOS

Arquitectura cliente-servidor, programación sockets C#

Descripción:

1.1 Los elementos básicos de la arquitectura cliente servidor

1.2 Comunicación de procesos remotos: direcciones IP, formato red/host, DNS.

Se describe el formato de las direcciones IP y cómo las podemos obtener a partir del nombre de una máquina, en concreto, se describen las llamadas a sistema que permiten la transformación. De forma muy descriptiva se habla de DNS que es el servicio que en realidad proporciona la traducción nombre máquina-IP

Finalmente se habla de la problemática de los diferentes formatos que pueden tener las arquitecturas de un computador (Little endian y big endian) y la necesidad de definir un formato red, formato en que han de viajar los datos.

1.3 Protocolo de comunicación orientado a conexión TCP. Llamadas a sistema/librería del lenguaje necesarias para implementar un cliente-servidor que utilice el protocolo TCP.

1.4 Protocolo de comunicación orientado a mensaje: UDP. Llamadas a sistema/librería del lenguaje necesarias para implementar un cliente-servidor que utilice el protocolo UDP.

Actividades vinculadas:

Actividades 1 y 3.

Dedicación: 30h

Grupo pequeño/Laboratorio: 8h 36m

Actividades dirigidas: 4h 30m

Aprendizaje autónomo: 16h 54m

Elementos básicos de la programación de bases de datos multinivel en C#

Descripción:

2.1 Elementos básicos de la arquitectura de bases tipo SQLite.

2.2 Creación de tablas multinivel. Definir las tablas de una base de datos y sus relaciones o índices.

2.3 Sentencias SQL complejas. Definir consultas a una base de datos con el lenguaje SQL.

Actividades vinculadas:

Actividades 1 y 3.

Dedicación: 10h

Grupo pequeño/Laboratorio: 2h 48m

Actividades dirigidas: 2h

Aprendizaje autónomo: 5h 12m



Sistema operativo Unix/Linux

Descripción:

3.1 Sistemas Operativo Multiusuario, Concurrentes.

3.2 Entorno de usuario: shell. Mandos básicos para la gestión de procesos y de ficheros desde la shell. Concepto de proceso y de fichero. Identificadores de Procesos y de ficheros. Ejecución de procesos secuencial y concurrente. Matar procesos. Protección de ficheros.

3.3 Entorno de programador: llamadas a sistema. Esquema de un programa en C que utilice llamadas a sistema. Compilación y ejecución de programas (secuencial y concurrente)

3.4 Llamadas a sistema de ficheros. Cómo crear, destruir, consultar y modificar un fichero con las llamadas a sistema. Comunicación asíncrona con ficheros.

Actividades vinculadas:

Actividades 1 y 3.

Dedicación: 30h

Grupo pequeño/Laboratorio: 8h 36m

Actividades dirigidas: 16h 54m

Actividades dirigidas: 4h 30m

Concurrencia Unix/Linux

Descripción:

4.1 Procesos. Jerarquía de procesos en Unix. Cómo crear, destruir procesos. Cambiar la imagen de un proceso. Comunicación de procesos padres-hijos síncrona. Concurrencia con procesos.

4.2 Threads. Concepto de thread. Cómo crear, destruir threads. Comunicación entre threads, memoria compartida. Concurrencia con threads.

Actividades vinculadas:

Actividades 2 y 3.

Dedicación: 30h

Grupo pequeño/Laboratorio: 8h 36m

Actividades dirigidas: 4h 30m

Aprendizaje autónomo: 16h 54m

Sockets en Unix/Linux

Descripción:

4.1 Sockets TCP. Llamadas a sistema para implementar un cliente-servidor que se comunica mediante sockets TCP.

4.2 Sockets UDP. Llamadas a sistema para implementar un cliente-servidor que se comunica mediante sockets UDP.

Actividades vinculadas:

Actividades 2 y 3.

Dedicación: 30h

Grupo mediano/Prácticas: 8h 36m

Actividades dirigidas: 4h 30m

Aprendizaje autónomo: 16h 54m



Acceso a Bases de datos MySQL desde un servidor Unix/Linux

Descripción:

- 4.1 Elementos básicos de la arquitectura de bases tipo MySQL.
- 4.2 Instalación MySQL en un servidor Unix.
- 4.3 SQL en un entorno Unix: Creación, consulta, modificación.

Actividades vinculadas:

Actividades 1 i 3.

Dedicación: 20h

Grupo pequeño/Laboratorio: 5h 48m

Actividades dirigidas: 3h

Aprendizaje autónomo: 11h 12m

ACTIVIDADES

PUZZLE: LINUX, BASES DE DATOS Y SOCKETS

Descripción:

Las actividades dirigidas y de aprendizaje autónomo consistirán en el estudio de material de autoaprendizaje, realización de ejercicios individuales y realización de ejercicios en pequeños grupos.

Las sesiones de clase se dedicarán a:

- Resolución de dudas del material de autoestudio, en pequeños grupos.
- Resolución de las dudas más frecuentes por parte del profesor.
- Ejercicios individuales y en pequeños grupos.

Objetivos específicos:

Al finalizar esta actividad los/las estudiantes serán capaces de:

- Utilizar los sistemas operativos de la familia de Linux a modo de usuario avanzado, tanto desde la línea de mandos como desde la interfaz de ventanas.
- Utilizar las llamadas a sistema del estándar POSIX desde programas escritos en C.
- Explicar el significado de los conceptos de socket y signal (interrupción software).
- Definir los permisos de acceso a un fichero y los dominios de protección.
- Crear y acceder a Bases de Datos multinivel.

Material:

Es el siguiente:

- Material de autoaprendizaje con los contenidos del tema.
- Enunciados de ejercicios individuales y en grupo.
- Plan detallado de actividades y entregas.

Todo el material estará disponible a través de Atenea.

Entregable:

La actividad tiene asignadas una serie de entregas individuales y en grupo (al menos una entrega por semana). Sobre la base de estas entregas se articularán los procesos de retroalimentación pertinentes.

La realización a tiempo de al menos el 80% de las entregas del curso será condición necesaria para superar la asignatura.

Algunas de las entregas son, de hecho, pruebas de conocimientos básicos puntuables.

Dedicación: 34h 30m

Grupo pequeño/Laboratorio: 9h

Actividades dirigidas: 4h 30m

Aprendizaje autónomo: 21h



PRÁCTICA GUIADA SOBRE THREADS EN LINUX Y C#

Descripción:

Las actividades dirigidas y de aprendizaje autónomo consistirán en el estudio de material de autoaprendizaje, realización de ejercicios individuales y realización de ejercicios en pequeños grupos.

Las sesiones de clase se dedicarán a:

- Resolución de dudas del material de autoestudio en pequeños grupos.
- Resolución de las dudas más frecuentes por parte del profesor.
- Ejercicios individuales y en pequeños grupos.

Objetivos específicos:

Al finalizar esta actividad, los/las estudiantes serán capaces de:

- Explicar el significado de los conceptos de proceso, programa y thread (hilo de ejecución).
- Determinar, a partir de los conceptos de proceso y thread, qué modelo de programación paralela es más adecuado para resolver un problema.
- Utilizar las llamadas a sistema de estándar POSIX desde programas escritos en C.

Además, en el contexto de esta actividad, los/las estudiantes desarrollarán algunas de las competencias genéricas identificadas antes en esta ficha (en particular, comunicación oral)

Material:

Es el siguiente:

- Material de autoaprendizaje con los contenidos del tema.
- Enunciados de ejercicios individuales y en grupo.
- Plan detallado de actividades y entregas.

Todo el material estará disponible a través de Atenea.

Entregable:

La actividad tiene asignadas una serie de entregas individuales y en grupo (al menos una entrega por semana). Sobre la base de estas entregas se articularán los procesos de retroalimentación pertinentes.

La realización a tiempo de al menos el 80% de las entregas del curso será condición necesaria para superar la asignatura.

Algunas de las entregas son, de hecho, pruebas de conocimientos básicos puntuables.

Dedicación: 11h 30m

Grupo pequeño/Laboratorio: 3h

Actividades dirigidas: 1h 30m

Aprendizaje autónomo: 7h



PROYECTO

Descripción:

En esta actividad, los estudiantes deberán hacer un proyecto de programación en equipos de tres o cuatro. Se utiliza la metodología de aprendizaje basado en proyectos, de manera que los/las estudiantes deben aprender de manera autónoma los temas que necesitan para alcanzar los objetivos del proyecto.

Las actividades dirigidas y de aprendizaje autónomo consisten básicamente en:

- Estudio de material de autoaprendizaje.
- Realización de ejercicios individuales y en grupo.
- Realización de las tareas individuales del proyecto.
- Reuniones de grupo para realizar las tareas del proyecto.
- Realización del diseño y planificación de los diferentes prototipos del proyecto.

Las actividades que se harán en la sesiones de clase son:

- Resolución de dudas del trabajo semanal en pequeños grupos.
- Resolución de las dudas más frecuentes por parte del profesor.
- Algunas sesiones expositivas sobre aspectos clave.
- Ejercicios individuales y en pequeños grupos.
- Realización de las tareas individuales del proyecto.
- Reuniones de grupo para realizar tarea del proyecto.

En esta actividad se dedicará una atención especial a la presentación escrita y oral del trabajo realizado por los equipos.

Objetivos específicos:

Al acabar esta actividad, los estudiantes serán capaces de:

- Explicar el significado de los conceptos entrada/salida siguientes: dispositivo, canal, fichero, canales estándar, redirección, acceso secuencial, acceso directo, compartición.
- Utilizar los sistemas operativos de la familia de Linux a modo de usuario, avanzando tanto desde la línea de mandos como desde la interfaz de ventanas.
- Utilizar las llamadas a sistema del estándar POSIX desde programas escritos en C.
- Saber resolver una aplicación de tipo cliente/servidor usando sockets, tanto con el protocolo TCP como UDP.
- Determinar, a partir los conceptos de proceso y thread, qué modelo de programación paralela es más adecuado para resolver un problema.
- Crear i acceder a Bases de Datos multinivel.

Además, en el contexto de esta actividad, los/las estudiantes desarrollarán las competencias genéricas identificadas antes en esta ficha.

Material:

Es el siguiente:

- Material de autoaprendizaje con los contenidos del tema.
- El entorno de programación Microsoft Visual Studio, Eclipse i Linux.
- Enunciados de ejercicios individuales y en grupo.
- Plan detallado de actividades y entregas.

Todo el material estará disponible a través de Atenea.

Entregable:

La actividad tiene asignadas una serie de entregas individuales y en grupo (al menos una entrega por semana). Sobre la base de estas entregas se articularán los procesos de retroalimentación pertinentes.

La realización a tiempo de al menos el 80% de las entregas del curso será condición necesaria para superar la asignatura.

Algunas de las entregas son, de hecho, pruebas de conocimientos básicos puntuables.

La actividad también tiene tres entregas importantes relacionadas con el resultado del proyecto y que contribuyen con un total de un 50% a la calificación final de la asignatura.

Dedicación: 104h

Grupo pequeño/Laboratorio: 31h

Actividades dirigidas: 17h

Aprendizaje autónomo: 56h



SISTEMA DE CALIFICACIÓN

La calificación se determinará a partir de estos componentes:

- Trabajos (10%)
- Exámenes (30%)
- Proyecto (50%)
- Actitud y participación (10%)

NORMAS PARA LA REALIZACIÓN DE LAS PRUEBAS.

Para superar la asignatura, será condición necesaria realizar a tiempo al menos el 80% de las entregas del curso.

También será condición necesaria superar todos los conocimientos básicos (o todos salvo uno).

El profesor proporcionará al/ a la estudiante el listado de conocimientos básicos al comenzar el curso.

BIBLIOGRAFÍA

Básica:

- Stevens, W.R. Advanced programming in unix environment. Ed. Addison Wesley, 1992. ISBN 0201433079.

Complementaria:

- Nemeth, E. UNIX system administration handbook. 2a. Upper Saddle River: Ed. Prentice Hall PTR, 2001. ISBN 0130206016.
- Kernighan, B.W. El entorno de programación Unix. Mèxic: Ed. Prentice-Hall Hispanoamericana, 1987. ISBN 9688800678.
- Tanenbaum, A.S. Modern operating systems. Englewood Cliffs: Ed. Prentice-Hall International, 1992. ISBN 0135881870.
- Márquez García, Fancisco M. UNIX: programación avanzada. 2a ed. Madrid: Ed. Ra-Ma, 1996. ISBN 8478972390.
- Silberschatz, Abraham; Galvin, Peter Baer; Gagne, G. Operating system concepts. 8a ed. Hoboken: Ed. John Wiley & Sons, 2010. ISBN 9780470233993.