



# Arquitectura de Computadores (25671) Enginyer de Telecomunicació (550) (1<sup>er</sup> Curso – 2<sup>o</sup> Semestre, 9 créditos)

**Curso 2007/08**

## **HORARIO:**

Martes	16-18h	Teoría	Prof. Eduardo César
Miércoles	15-17h	Teoría y Problemas	Prof. J.C. Moure
Jueves	15-16h	Problemas	Prof. Eduardo César
Lunes/Jueves	Tarde	Laboratorio	Paula Fritzsche

## **PROGRAMA: Sistemas Operativos**

1. Definiciones Previas (1h)
  - Estructura general del procesador + modo protegido/usuario
  - Ciclo de ejecución
  - Interrupciones (ideas + causas)
2. Antecedentes y concepto de los Sistemas Operativos (2h)
  - Concepto de Sistema Operativo
  - Evolución histórica.
  - Estructura modular del SO
  - Concepto de Kernel.
3. Introducción al SO UNIX (nivel usuario) (6h -4+2-)
  - Origen de UNIX
  - Características más importantes (estructura de directorios, estructura de los comandos, interficie con los usuarios)
  - Comandos básicos y redirección de entrada/salida
  - Archivos de comandos y comandos avanzados (variables, aritmética, estructuras de control, awk)
4. Procesos y Planificación (5h -4+1-)
  - Concepto de proceso.
  - Estados de un proceso.
  - Estructuras de datos asociadas.
  - Concepto de thread (hilo)
  - Cambio de contexto vs. Cambio de modo
  - Concepto de planificación. Planificación a corto plazo.
  - Criterios de planificación y criterios de evaluación.
  - Políticas de planificación.
    - a. No apropiativas: FCFS, SPN.
    - b. Apropiativas: SRT, RR, Prioridad (colas multinivel)
5. Sistema operativo UNIX (nivel programador) (6h -4+2-)
  - Estructuras de datos del SO UNIX para la gestión de la E/S
  - Llamadas al sistema para la gestión de archivos.
  - Llamadas al sistema para la gestión de procesos.
  - Llamadas al sistema para la gestión de threads.

6. Concurrencia (5h -3+2-)
  - Descripción del problema de la sección crítica.
  - Sincronización por semáforos.
  - Sincronización por mensajes.
  - Llamadas al sistema para la gestión de sockets.
  - Llamadas al sistema para la gestión de mutex y variables condicionales.
  - Deadlock.
7. Sistema de Archivos (4h -3+1-)
  - Conceptos básicos.
  - Métodos de acceso a la información.
  - Directorios.
  - Gestión de almacenamiento secundario.
    - a. Bloques.
    - b. Espacio libre (bit maps, listas, índices).
    - c. Asignación de espacio (continua, encadenada, indexada, FAT, I-nodos, NTFS).
8. Gestión de E/S (4h -3+1-)
  - Relación con el sistema de archivos.
  - Técnicas de E/S.
  - Diseño del sistema de E/S.
  - Estructura de un driver.
  - Buffering y cache de disco.
  - Comentario sobre planificación de disco.
    - a. Políticas orientadas al usuario.
    - b. Políticas orientadas al sistema.
9. Gestión de Memoria (4h -3+1-)
  - Conceptos básicos.
  - Memoria Principal.
    - Paginación, Segmentación, Segmentación Paginada
  - Memoria Virtual.
    - a. Concepto y requerimientos. Políticas del SO.
    - b. Políticas de reemplazo: Óptima, FIFO, LRU, 2ª oportunidad
    - c. Hiperpaginación, gestión del conjunto residente.
10. Seguridad (3h)
  - Amenazas a la seguridad.
  - Protección de acceso.
  - Protección de archivos.
  - Virus y afines.
  - Comunicaciones.
11. SO Tiempo Real i para dispositivos móviles (7h -5+2-)
  - Motivación y conceptos (Hard RT, Soft RT, tareas periódicas)
  - Requisitos de un SO tiempo real.
  - Principales características que permiten TR.
  - Planificación en tiempo real.
    - a. Métodos básicos.
    - b. Planificación por plazos.
    - c. Planificación monótona de frecuencia.
  - Consideraciones sobre ahorro de energía y uso de recursos.

## PROGRAMA: Arquitectura de Computadores

1. Introducción (1,5h)
  - Evolución histórica de los computadores. Ley de Moore
  - Rendimiento y coste: definiciones, ecuación de rendimiento
  - Principios cuantitativos de diseño de computadores: paralelismo, localidad y análisis del caso más frecuente
2. Paralelismo a Nivel de Instrucciones (ILP) (6,5h)
  - Paralelismo: Dependencia de Datos y de Control. Camino Crítico
  - Paralelismo y Recursos de Cómputo. Límite de Rendimiento
  - Ejecución Segmentada de instrucciones (*Pipeline*)
  - Anticipación de Datos y Predicción de Saltos
  - Procesadores VLIW o EPIC.
  - Técnicas estáticas para aumentar el paralelismo a nivel de instrucciones (ILP)
  - Técnicas dinámicas para aumentar el ILP: planificación dinámica y renombrado de registros, predicción de saltos y ejecución especulativa.
3. Sistema de Almacenamiento e Interconexión (7h)
  - Capacidad, Localidad y Ancho de Banda.
  - Paralelismo en el acceso a los datos: solapamiento y anticipación
  - Tecnología de memoria principal y de almacenamiento secundario
  - Tecnología de los elementos de interconexión
  - Jerarquía de memorias e interconexión: motivación y evaluación
  - Automatización de la jerarquía: Memoria Caché y Memoria Virtual
  - Estrategias S/W para mejorar el rendimiento de la memoria
4. Repertorio de Instrucciones y Procesamiento Digital de Señales (2h)
  - Influencia del repertorio de instrucciones en el rendimiento
  - Tipos de repertorio de instrucciones: CISC, RISC, DSP, VLIW (EPIC)
  - Características de las aplicaciones DSP
  - Repertorio de instrucciones DSP
  - Arquitectura de procesadores DSP
5. Paralelismo a Nivel de *Threads* y de Tareas (TLP) (7h)
  - Paralelización a nivel de tareas. Rendimiento de las aplicaciones paralelas
  - Comunicación y sincronización entre tareas. Solapamiento entre cómputo y comunicaciones.
  - Programación de las aplicaciones paralelas: MPI y OpenMP.
  - Ejemplos de algoritmos paralelos
  - Arquitecturas de multiprocesadores: memoria compartida y distribuida
6. Sistemas Empotrados y Servidores (2h)
  - Fiabilidad y Disponibilidad. Consumo y Potencia
  - Características de los sistemas empotrados
  - Características de los Sistemas Servidores. Modelo de Colas.
  - Servidores de Almacenamiento y de Entrada/Salida

## **BIBLIOGRAFIA:**

### **Básica (Sistemas Operativos)**

1. **Sistemas Operativos.** 4ª edición. W. Stallings. Prentice Hall.

### **Básica (Arquitectura)**

2. **Computer Architecture. A Quantitative Approach. Third Edition.**  
J. Hennesy & D. Patterson. Morgan Kaufmann (Elsevier), 2003.

### **Recomendada (Sistemas Operativos)**

3. **Sistemas Operativos. Una visión aplicada.** Jesús Carretero Pérez, Pedro de Miguel Anasagasti, Félix García Caballería, Fernando Pérez Costoya. Mc Graw Hill.
4. **Operating Systems Concepts.** Fifth edition. Silberschatz, Galvin. Wiley.

### **Recomendada (Arquitectura)**

5. **Organización y Arquitectura de Computadores.** 5ª Edición. W. Stallings. Prentice Hall, 2000
6. **Computer Organization and Design. The Hardware/Software Interface, Third Edition,** D. Patterson & J. Hennesy. Morgan Kauffman (Elsevier), 2005.
7. **Parallel Programming in C with MPI and OpenMP.** Michael J. Quinn. McGraw Hill, 2004
8. **Embedded Computing: A VLIW approach to architecture compilers and tools.** Joseph A. Fisher, Paolo Faraboschi, Cliff Young. Morgan Kauffman (Elsevier), 2005.
9. **Digital Signal Processing and the Microcontroller.** Dale Grover & John R. Deller. Prentice Hall, 1999
10. **BDTI Publications and Presentations.**  
[http://www.bdti.com/articles/info\\_articles.htm](http://www.bdti.com/articles/info_articles.htm)
11. **Embedded Systems Dictionary.** Jack Ganssle and [Michael Barr](#).  
<http://www.netrino.com/Books/Dictionary/contents.html>
12. **The Scientist and Engineer's Guide to Digital Signal Processing (chapter 28).** *Steven W. Smith, Ph.D. California Technical Publishing,* Accesible por internet en: <http://www.dspguide.com/pdfbook.htm>
13. **Computers as Components: principles of Embedded Computer system design,** Wayne Wolf, Morgan Kauffman, 2005.

## MÉTODO DE EVALUACIÓN

- o **NO** hay examen final: se realizará una evaluación continua.
- o La asignatura se divide en dos tipos de contenidos temáticos: Sistemas Operativos (SO) y Arquitectura de Computadores (AC).
- o De forma **obligatoria** cada alumno deberá realizar individualmente **3 Pruebas Parciales (para cada una de las partes de la asignatura)**.
- o De forma **obligatoria** cada alumno deberá realizar prácticas de laboratorio en grupos de 2 personas.
- o Durante las sesiones de teoría se propondrán problemas para realizar y entregar durante la clase. Los alumnos podrán resolverlo en parejas y entregarlo por escrito. La entrega de problemas NUNCA penaliza (ver ejemplo abajo)
- o Las siguientes condiciones son **necesarias** para aprobar la asignatura:
  - o En cada una de las 3 pruebas parciales individuales se debe sacar una nota  $\geq 4$  sobre 10. Esto debe cumplirse tanto para la materia de SO como para la de AC.
  - o Las prácticas deben **aprobarse** con una nota  $\geq 5$  sobre 10.
  - o La nota final de cada una de las partes (SO y AC), calculada como se indica adelante, debe ser  $\geq 5$  sobre 10.
- o En 2ª convocatoria se pueden recuperar por separado cada una de las pruebas parciales y las prácticas. No se guardan notas ni prácticas de un curso a otro.

### Para cada materia (SO ó AC) la nota parcial se calcula como:

Nota de los problemas:	entre 0% y 40% (nunca penaliza)
Nota de las pruebas parciales:	entre un 60% y un 100%

Suponer que un alumno obtiene un 4 sobre 10 en las tres pruebas parciales.  
**Ejemplo 1:** se han obtenido 2 puntos sobre 4 en los problemas, así que el alumno se evalúa en las pruebas parciales sobre 8 puntos. La nota final sería de  $2 + 4 \times 8/10 = 5,2$  (APROB)  
**Ejemplo 2:** con 0 puntos de problemas, el alumno anterior se evaluaría en las pruebas parciales sobre 10 puntos. Nota= 4 (SUSP)  
**Ejemplo 3:** con 4 puntos de problemas y un 5 en las pruebas parciales, la nota final sería de  $4 + 5 \times 6/10 = 7,0$  (NOTABLE)

### La CALIFICACIÓN FINAL se calcula como

Nota de Sistemas Operativos:	<b>50%</b>
Nota de Arquitectura de Computadores:	<b>35%</b>
Nota de Prácticas:	<b>15%</b>

	<b>PROFESORES</b>	<b>E-Mail</b>
<i>TEORÍA :</i>	Eduardo César Juan C. Moure	Eduardo.Cesar@uab.es JuanCarlos.Moure@uab.es
<i>PRÁCTICAS :</i>	Paula Fritzsche	paula@aomail.uab.es

**Este documento y el resto de documentación de la asignatura se publicarán en el Campus Virtual**

## **PRÀCTICAS**

Las prácticas se realizarán sobre el sistema operativo Linux, y constarán de un conjunto de ejercicios que cubrirán los siguientes temas:

- Programación de archivos de comandos en UNIX (1 sesión)
- Creación de procesos (1 sesión)
- Comunicación mediante paso de mensajes (sockets) (1 sesión)
- Creación de Threads (procesos ligeros) (1 sesión)
- Comunicación mediante memoria compartida (1 sesión)

En el caso de las prácticas la evaluación también es continua, teniéndose en cuenta la calidad de las soluciones propuestas y también su entrega puntual. Durante las sesiones de laboratorio se contará con la presencia de un profesor de prácticas que será responsable de guiar la realización pero también de la evaluación de los diferentes ejercicios.