Palma, 26 de octubre de 2008

A la Comisión para la Elaboración y Diseño del título de grado de Ingeniería Informática

Adjunto la **propuesta de mejora de los descriptores** de las asignaturas **"Sistemas Operativos" y "Ampliación de Sistemas Operativos"** para el nuevo título de Ingeniería Informática. Aunque no existen grandes diferencias con los descriptores actuales en el último borrador de la propuesta, creo que mi propuesta agrega detalles y se ajusta mejor a los módulos –la mayoría "nucleares"-- propuestos por la ACM.

También propongo el **cambio de los los nombres de la asignatura a "Sistemas Operativos 1" y "Sistemas Operativos 2"** respectivamente. La justificación de este cambio se encuentra al final de la justificación de las modificaciones propuestas para la actual "Ampliación de Sistemas Operativos".

Al final del documento hago unos comentarios y sugerencias para las asignaturas **"Diseño de sistemas distribuidos"** y **"Sistemas operativos distribuidos"**. Aunque conozco algo del tema por mi tesis doctoral, no soy un especialista del mismo, ni conozco los objetivos o ideas detrás de ambas asignaturas. Sin embargo me parece que sus descripciones no son claras y en algunos casos contradictorias.

Espero que sirva de ayuda.

Ricardo Galli Granada

DMI

DNI: xxxxxxxxx

Sistemas Operativos 1 (Sistemas Operativos)

Propuesta detallada (en negrita lo que podrían ser los descriptores breves)

- **Introducción: objetivos, funcionalidad, aspectos de diseño**
  - Papel y objetivos de los SO.
  - Historia del desarrollo.
  - Funcionalidad de sistemas típicos.
  - Aspectos de diseño: eficiencia, fiabilidad, flexibilidad, portabilidad, seguridad, compatibilidad.
- **Principios fundamentales, estructuras, abstracciones**
  - Estructuras: monolíticos, micronúcleos, modular.
  - Abstracciones, procesos, recursos.
  - Requerimientos y evolución del software y hardware.
  - Organización de dispositivos.
  - Interrupciones: mecanismos e implementación.
  - Conceptos de modo sistema, modo usuario y seguridad. Transiciones.
- **Concurrencia**
  - Estados y diagramas de estado de procesos.
  - Estructuras (cola de listos, de bloqueados, suspendidos, etc.).
  - Despacho y cambios de contexto. El papel de las interrupciones.
  - Ventajas y desventajas de ejecución concurrente.
  - El problema de la exclusión mutua. Diferentes soluciones.
  - Interbloqueo: causas, condiciones, prevención
  - Modelos y mecanismos: semáforos, monitores, variables de condición, sincronización.
  - Modelos típicos: productor-consumidor, lectores-escritores.
  - Aspectos y problemas de multiprocesadores (*spinlocks*, coherencia, *re-entrancia*)
- **Planificación de procesador**
  - Apropiativos y no apropiativos.
  - Planificación y políticas.
  - Procesos e hilos.
  - Estructuras de datos, colas múltiples, prioridades dinámicas.
  - Latencias, tiempos de respuestas.

Justificación

Los temas de Sistemas Operativos cubren cuatro módulos –en el mismo orden-- de los módulos nucleares definidos en la última versión disponible de la ACM: http://wiki.acm.org/cs2001/index.php?title=Operating_systems (*OS/OverviewOfOperatingSystems*, *OS/OperatingSystemPrinciples*, *OS/Concurrency* y *OS/SchedulingAndDispatch*).

Estos temas son prácticamente iguales a los que se dan en la actual *Sistemes Operatius* (4512) que ya está

bastante ajustada a los módulos definidos por la ACM (tampoco es un gran mérito del profesor actual, los libros de texto de calidad ya están ajustados y ordenados a dichos módulos).

El objetivo fundamental de esta asignatura es que los alumnos conozcan la evolución y diseño de los sistemas operativos desde la visión de ingeniería de software y las influencias mutuas con el desarrollo y evolución del hardware.

La segunda parte de la asignatura presenta los problemas fundamentales de concurrencia del desarrollo de un sistema operativo y de la multiprogramación en general como así también los algoritmos y mecanismos básicos para resolverlas (fundamentalmente basadas en algoritmos de memoria compartida). En todo el estudio es fundamental el concepto de procesos, la *ejecución en paralelo* e *intercalación*, es por ello que el módulo de planificación de procesador ayuda a la comprensión en detalle de la implementación y los problemas derivados.

La propuesta actual es muy similar, esta sólo desarrolla un poco más y elimina el descriptor “Ilustración de conceptos sobre SO actuales” ya que además que es un tema transversal a la asignatura, en general los conceptos son fundamentales e independientes, salvo la planificación de procesador donde sí hay variaciones en la implementación e interfaces.

Sistemas Operativos 2 (Ampliación de Sistemas Operativos)

Propuesta detallada (en negrita lo que podrían ser los descriptores breves)

- **Gestión de memoria**
  - Introducción a memoria física y hardware de gestión de memoria.
  - Evolución y aspectos de diseño.
  - Seguridad.
  - Paginación y memoria virtual.
  - Conjuntos de trabajo.
  - Cache y tablas de *traducción adelantada* (TLB).
- **Gestión de dispositivos**
  - Abstracción de los diferentes tipos (caracteres, bloques).
  - Tipos de acceso (programada, por interrupciones, acceso directo a memoria).
  - Planificación de E/S en dispositivos de bloque.
- **Sistemas de ficheros**
  - Gestión espacio libre asignado.
  - Datos, metadatos, organización, *buffering*, *journaling*.
  - Sistemas de tablas de asignación y nodos índices.
  - Directorios, denominación, búsqueda, cache.
  - Mapeado en memoria, cache de páginas, *serialización*.

Justificación

Como en Sistemas Operativos 1, esta propuesta se ajusta a los módulos de la ACM: *OS/MemoryManagement*, *OS/DeviceManagement* y *OS/FileSystems*. Los dos últimos son opcionales en la propuesta de la ACM, aunque son relevantes para un título de grado de ingeniería.

Gestión de dispositivos da la visión general de la estructura de lo que suele ser aproximadamente el 80% del código de sistemas operativos modernos (*device drivers*). El estudio de la planificación de E/S es muy pedagógica para mostrar las enormes diferencias y ventajas que se obtienen al ordenar las colas de E/S de forma de minimizar los tiempos de latencia generado por el movimiento físico de las cabezas lectoras-grabadoras de los discos duros. No hace falta insistir demasiado en las abstracciones de otros tipos de dispositivos ya que en general se tratan como dispositivos de caracteres y sus diferencias y complejidad variable depende casi exclusivamente del diseño del hardware (por ello tiene menos descriptores que el original de la ACM).

En el borrador actual se diferencia entre memoria principal (RAM) y secundaria (disco). Esta diferenciación prácticamente no tiene sentido ya que los sistemas modernos de memoria virtual usan paginación, cuya "traslación" a "memoria secundaria" es muy simple y no tiene sentido estudiarlas por separado.

En el mismo documento hay un error importante al asociar "memoria virtual" a la "memoria principal". Se denomina "virtual" justamente porque es la unificación de ambas que se presentan como una única memoria e interfaz similar a la memoria RAM (cuando en realidad tiene partes o conjuntos almacenadas en la secundaria).

El descriptor actual "Ilustración de conceptos sobre SO actuales" ya que es un tema transversal en toda la asignatura, por lo que no tiene sentido ponerla como un descriptor independiente.

Estos temas propuestos son prácticamente iguales a los que se dan en la actual *Ampliació de Sistemes Operatius* (4520) que ya está bastante ajustada a los módulos definidos por la ACM, salvo la unificación actual de "memoria virtual" en un único descriptor que en la asignatura actual están separados (y generan una dificultad al intentar enseñarlos independientemente).

Como se puede observar, los temas de "Sistemas Operativos 2" ("Ampliación de Sistemas Operativos" en la propuesta actual) son diferentes y en muchos sentidos independientes de los de "Sistemas Operativos 1", por lo que considero que "ampliación" en el nombre de la asignatura es confuso y no se ajusta a la realidad. Por ello es que **propongo denominarles "Sistemas Operativos 1" y "Sistemas Operativos 2"**.

Nota sobre el bloque de competencias

En el "Bloque de competencias específicas (grupo 3)" (Sistemas Operativos) se comete un error importante. Dice:

> *Diseñar e implementar **aplicaciones monoproceso** basadas en los servicios del SO, seleccionando los más adecuados a cada caso.*

El error es que los sistemas operativos modernos presentan una abstracción e interfaz de [multi] programación uniforme e independiente del número de procesadores del sistema. Además el estándar de los sistemas operativos modernos es que todos ofrecen mecanismos de multiprogramación (salvo casos de empotrados y controladores muy específicos) por lo que hablar de "monoproceso" cualquiera sea su acepción (procesadores o números de procesos activos) es un anacronismo contradictorio y sin valor académico añadido.

Las técnicas de resolución de problemas de concurrencia son independientes de que sean mono o multiprocesador, ya que los problemas de "intercalación exclusiva" y "ejecución paralela" son equivalentes. Salvo la excepción de la necesidad mecanismos internos del sistema operativo –relativamente sencillos-- para asegurar la coherencia de cache (invalidación de caches) y acceso atómico a variables "globales" de sincronización (*spinlocks*). Pero estos métodos están incluidos en todos los sistemas operativos y de procesadores modernos, especialmente desde que las arquitecturas SMP y de múltiples núcleos se han convertido en un estándar incluso para el mercado doméstico.

Afortunadamente este error no ha influido en la propuesta de contenidos.

Sistemas operativos distribuidos (y Diseño de Sistemas Distribuidos)

Los descriptores no dejan claro si se trata de "sistemas distribuidos" o "sistemas *operativos* distribuidos" (ni qué tipos de *sistemas operativos* distribuidos se estudiarán, cliente-servidor o de memoria distribuida). Algún descriptor hace suponer que se trata de "sistemas operativos distribuidos" basados el "memoria compartida distribuida", o quizás de sistemas de "imagen única".

Globalmente no se puede distinguir las diferencias con "Diseño de sistemas distribuidos", donde se estudian temas como "distribución basada en objetos" que es una tercera alternativa implementada –por ejemplo-- por el sistema operativo E1 (previsto para 2005 aunque parece que nunca fue publicado) o sistemas de *clustering* similares basados en distribución como Mosix.

Aún suponiendo que se trata de temas específicos de sistemas operativos distribuidos, los descriptores son confusos o de granularidad no comparables (algunos de ellos parecen ser sacados del libro "Distributed Systems: Principles and Paradigms" de Andrew S. Tanenbaum y Maarten van Steen). Por ejemplo "servicios de sincronización de reloj", no se sabe si se refiere a algoritmos, técnicas o "servicios" propiamente dicho. En realidad es parte de un problema más general que es "sincronización" que incluye problemas de reloj, *election*, transacciones y exclusión mutua distribuida.

El desarrollo de *sistemas operativos distribuidos "transparentes"* ha perdido mucho empuje en los últimos años (incluso sistemas tan emblemáticos como Amoeba están parados desde 2001). En general se debe a las ineficiencias de los sistemas basados en memoria compartida, y que los sistemas de intercambio y migración de objetos han sido reemplazados por sistemas más simples que obtienen mejores rendimientos con "sistemas operativos de red tradicionales" (sistemas de grid, *clusters "Beowulf"*, mecanismos de paso de mensajes, sistemas de multiprocesador con múltiples núcleos, virtualización, sistemas de ficheros y bases de datos distribuidas, etc.).

Creo que es un error dedicar una asignatura de 6 créditos a estudiar sistemas que ya no se siguen investigando ni desarrollando comercialmente. Es más útil estudiar temas más genéricos que complementen y completen a "sistemas distribuidos". Esta tendencia es patente y notables en los módulos *Net Centric Computing* de la ACM: http://wiki.acm.org/cs2001/index.php?title=Net_Centric_Computing

No soy experto en "sistemas distribuidos" (aunque en mi tesis doctoral diseñe e implementé el protocolo y algoritmo de replicación y consistencia de memoria compartida y exclusión mutua distribuida), tampoco conozco los objetivos de ambas asignaturas. Pero en el borrador no están claros los objetivos, ni las diferencias con "Diseño de sistemas distribuidos", ni por qué se denomina "sistema operativo distribuido" cuando quizás sea más acertado sólo "sistemas distribuidos".

Creo que una opción más apropiada sería llamarles **"Sistemas Distribuidos 1"** y **"Sistemas Distribuidos 2"** y que entre ambas contengan los siguientes descriptores:

- Comunicaciones y protocolos.
- Comunicación de grupo y ordenamiento de mensajes.
- Llamadas a procedimientos/objetos remotos.
- Sistemas de paso de mensajes y modelos de memoria compartida.
- Migración de procesos, agentes.
- Problemas de nombramiento (o "nominación") de entidades, dispositivos móviles.
- Sincronización (exclusión mutua, *election*, reloj, transacciones, etc.).
- Modelos de consistencia y replicación.
- Seguridad.
- Alta disponibilidad. Tolerancia a fallos. Balanceo de carga.
- Sistemas de ficheros/objetos distribuidos.

Palma, 27 de octubre de 2008

A la Comisión para la Elaboración y Diseño del título de grado de Ingeniería Informática

Adjunto la **propuesta de mejora de los descriptores** de la asignatura **"Sistemas abiertos e interfaces de usuario"** para el nuevo título de Ingeniería Informática. Aunque se proponen cambios importantes en los descriptores, los objetivos de la asignatura me parecen correctos, *creo* que se en la propuesta no se distorsionan esos objetivos iniciales, sólo se generalizan y adecúan a un lenguaje más técnico y coherente con el usado en las propuestas de currículum de la ACM.

Espero que sirva de ayuda.

Ricardo Galli Granada

DMI

DNI: xxxxxxxx

Desarrollo web e interfaces de usuarios

- Tecnología web y estándares aplicables
- Principios de ingeniería web y sitios dinámicos basados en bases de datos
- Patrones y entornos (*frameworks*) de desarrollo web
- Lenguajes dinámicos e interpretados
- Seguridad: Inyección SQL, XSS, CSRF, sesiones, *click hijacking*...
- Diseño de interfaces de usuarios, AJAX, librerías y APIs (prototype, jQuery...)

Justificación

Cambio de nombre

En la versión actual la asignatura se denomina “Sistemas abiertos...”, que no tiene un significado académico o comercial preciso.

El término tiene orígenes comerciales, apareció en los '80 para denominar a aquellos sistemas operativos y aplicaciones que eran compatibles con Unix. Con el tiempo el término no adquirió un significado más preciso, más allá que es a veces usado comercialmente para indicar que respeta “estándares abiertos”.

Aunque se suponga que ese es el significado, “uso de estándares abiertos” explica muy poco el objetivo de la asignatura, que es básicamente “desarrollo web”. Por eso propongo que se mantengan los objetivos e ideas fundamentales pero que se cambie el nombre a “Desarrollo web...”, que además es más próximo y coherente con el módulo *WebOrganization* de la propuesta de la ACM para *Net Centric Computing* (http://wiki.acm.org/cs2001/index.php?title=Net_Centric_Computing).

La versión del borrador

El primer descriptor del borrador actual, “Introducción a la programación web (XML, HTML)”, es bastante desafortunado. Parece dar a entender que en el último curso de la ingeniería recién se encontrarán con HTML y XML. Además es contradictorio decir “introducción a la programación” y citar un par de lenguajes de etiquetado que no de programación.

“Programación web interactiva (scripting)” es también desafortunado ya que todas las aplicaciones web son casi por definición “interactivas”, quizás se pretendía decir “programación de sitios dinámicos”. Además confunde “programación interactiva” con “scripting”, siendo este último una forma bastante anacrónica –derivados de los “scripts” de shell en Unix-- de denominar a la programación con “lenguajes dinámicos”.

“Sistemas de pasarela entre aplicaciones web y SGBD ” y “Servidores web y servidores de aplicaciones ” son tecnologías y conceptos –respectivamente-- que pueden ser englobados más genéricamente por “tecnologías y principios de ingeniería web”.

El descriptor “El modelo de tres capas o Modelo Vista Controlador” es también desafortunado ya que da la impresión que recién en esta asignatura aprenderán este patrón de diseño y desarrollo, cuando probablemente significa que se enseñarán los *frameworks* de “moda” que ayudan y dan soporte al uso de este patrón para el desarrollo de aplicaciones web.

Aunque no es tan relevante, tampoco hay un consenso sobre el uso “laxo” del nombre MVC, ya que en la mayoría de los *frameworks* no se ajusta a la definición original del modelo MVC. Por ejemplo el “controlador tradicional” está repartido entre el servidor web y el tratamiento de las entradas y URLs, ambas no suelen ser parte importante de lo que expone el *framework* al desarrollador, por eso algunos prefieren llamarles “Model-Template-View” que se ajusta mejor a lo que hacen los *frameworks* genéricamente llamados MVC (*Model* es el modelo de datos, *Template* es la definición de las plantillas para la generación

de la salida a partir de los datos expuestos por el *View*, siendo este último el módulo donde reside la lógica de la aplicación).

En la propuesta actual se olvida un tema importante y que está causando muchos problemas por desconocimiento, falta de práctica o porque se lo toma como un "tema secundario", los aspectos y problemas típicos de seguridad en aplicaciones web.

Propuesta

En la propuesta se introducen descriptores que se ajustan a las definiciones de *NC/WebOrganization* y *NC/NetworkedApplications* de la ACM. Todos ellos incluyen y generalizan las tecnologías y conceptos de los descriptores actuales.

Además se agrega "Lenguajes dinámicos e interpretados" que permite dar una introducción a los lenguajes más habituales de programación web dinámicos (PHP, Python, Perl, Ruby, Javascript/ECMAScript. VBScript) e interpretados (que incluye a lenguajes de código intermedio como Java o C#). Además al ser genérico permitirá que el profesor elija para la prácticas el lenguaje y el *framework* que considere más adecuado para sus clases (o lo que sea lo más usado o actual dentro de cuatro o cinco años cuando se de esta asignatura por primera vez)

Se ha añadido también "Seguridad" ya que es importante que los alumnos conozcan los problemas y [peligrosas] consecuencias del desarrollo web.

En "Interfaz de usuario" se agregó "AJAX", también coherente con el descriptor "Web Services, Web 2.0, ajax" de la ACM (aunque Web 2.0 es un término difuso y que probablemente pase de moda rápidamente, seguramente web 2.0 desaparezca de la versión definitiva de la ACM). AJAX es un nombre genérico que se da al uso extenso de la característica de "conexión en segundo plano" del Javascript/ECMAScript para el desarrollo de interfaces de usuarios con interacciones similares a las aplicaciones tradicionales que se ejecutan completamente el ordenador del usuario.

También se añadió "librerías y APIs" porque es muy habitual que se usen estas bibliotecas que implementan abstracciones de alto nivel de la interfaz de usuario (manipulan el XHTML, CSS y DOM en el navegador) como el Prototype y jQuery y todas las de más alto nivel construidas sobre estas dos librerías básicas.

Hecha la propuesta, reconozco que **tres créditos son muy pocos para una asignatura de este tipo**. Lo adecuado sería tener al menos seis créditos, o el profesor tendrá que hacer esfuerzos monumentales para resumir, simplificar y transmitir claramente los conceptos, tecnologías y herramientas fundamentales.