

WUOLAH



jesusjimsa

www.wuolah.com/student/jesusjimsa



1377

Tema 1 - Estructuras de Sistemas Operativos.pdf

Resumen Tema 1



2º Sistemas Operativos



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
UGR - Universidad de Granada

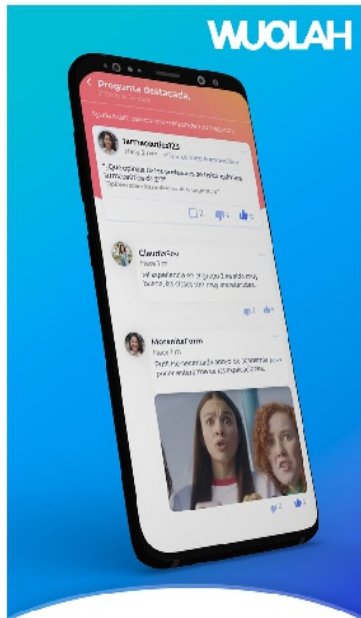
Tema 1: Estructuras de Sistemas Operativos

Un sistema operativo (SO) es un programa que controla la ejecución de programas de aplicación y actúa como una interfaz entre estas aplicaciones y el componente hardware del ordenador. Presenta tres objetivos principales:

- **Conveniencia** (interfaz usuario-ordenador): Presenta al usuario una máquina abstracta más fácil de programar, ocultando la complejidad del hardware y facilitando el acceso a utilidades, servicios y procesos. Para un usuario final (generalmente no programador) estos detalles tan complicados no son útiles y para un usuario programador, un lenguaje de alto nivel resulta más útil que un conjunto de enrevesadas instrucciones máquina. El SO proporciona servicios en las siguientes áreas:
 - Desarrollo de programas (editores, compiladores, depuradores...).
 - Ejecución de programas (carga, inicializa y repara los recursos de estos).
 - Acceso a dispositivos E/S (cada dispositivo requiere su propio conjunto de instrucciones que se ocultan al usuario).
 - Acceso controlado a ficheros.
 - Acceso al sistema (controla el acceso y uso de los recursos del sistema a diferentes usuarios).
 - Detección y respuesta a errores (hardware y software).
 - Contabilidad (estadísticas de uso y medida del tiempo de respuesta del sistema).
- **Eficiencia** (administrador de recursos): Un computador es un conjunto de procesos para el movimiento, almacenamiento y procesamiento de datos, y para ello posee un mecanismo de control que cubre estos dos aspectos:
 - Las funciones del SO actúan de la misma manera que el resto del software (programa ejecutados por el procesador).
 - El SO frecuentemente cede el control al procesador y depende del mismo para retomarlo.

Por lo tanto:

- El SO dirige al procesador en el uso de los recursos del sistema y en la temporización de la ejecución de otros programas.
- Una parte del SO está cargada en memoria principal (kernel/núcleo y, a veces, otras partes del mismo que se estén usando). El resto de la memoria la ocupan programas y datos de usuario.



Pregunta y responde todas las dudas con tus compañeros de clase a través de la app.



6 sitios web para descargar libros gratis de forma legal Fuente: noticiasuniversia.es

1. Editorial Traficantes de Sueños

La editorial Traficantes de Sueños y un símbolo de los movimientos sociales en Madrid. Todas sus publicaciones tienen copyleft (distribución de copias y versiones modificadas de una obra u otro trabajo) y pueden descargarse en formato .pdf.

2. Bubok

Bubok también es una editorial española que publica bajo demanda y brinda muchos libros gratuitos, y otros .pdf de pago, por deseo de sus autores. Busca en las diferentes categorías y encuentra lo que quieres.

3. Proyecto Gutenberg

Proyecto Gutenberg es una biblioteca de ebooks de dominio público gratuitos extremadamente conocida. En general ofrece libros en formato .epub y .mobi (Kindle).

4. Europeana

Europeana es la gran biblioteca digital de Europa y ofrece numerosos libros en formato digital y en general de dominio público. Todas las obras pueden encontrarse en cada uno de los idiomas oficiales de la Unión Europea.

5. Amazon

Amazon es una de las tiendas de venta online más extensas y respetadas a nivel mundial también ofrece los ebooks gratuitos para Kindle, principalmente las obras clásicas de la literatura en castellano, inglés y francés.

6. Anarres

La editorial argentina Anarres difunde la ideología anarquista de manera gratuita, en formato .pdf.

Las mejores aplicaciones Android para aprender inglés Fuente: noticiasuniversia.es

Duolingo

Esta app permite a sus usuarios aprender inglés, entre otros idiomas, de forma divertida y completamente gratis. La aplicación es interactiva y muy original, ya que funciona como un juego en el que perderás vidas por cada respuesta incorrecta, mientras que los aciertos te reportarán puntos con los que subir de nivel.

Busuu

Quizás, lo mejor de Busuu sea su inmensa comunidad formada por personas nativas con las que practicar tu inglés. Los usuarios también pueden acceder a prácticas de audición, lectura, escritura y conversación con las que poner fin a esa asignatura pendiente con los idiomas.

Babbel

La sencillez y éxito de Babbel reside en

la categorización de un máximo de 3.000 palabras en diferentes temáticas. Además de la posibilidad de ampliar vocabulario, la aplicación permite mejorar la pronunciación mediante diversos ejercicios prácticos.

Voxy

Especialmente destinada a quienes estén preparando las pruebas de certificación de nivel de inglés. Voxy ofrece ayuda online de tutores nativos y ejercicios para mejorar tu fluidez y pronunciación en tiempo récord.

British Council App

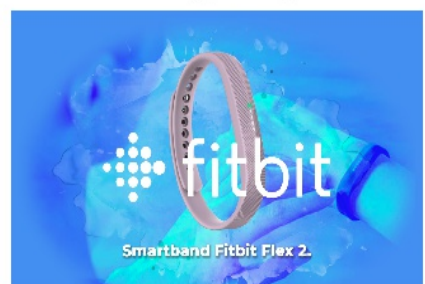
El Instituto Británico ofrece también diversas aplicaciones para aprender inglés según el nivel. El prestigio académico de la institución es una garantía de la calidad de las herramientas.

WUOLAH GIVEAWAY



Consola PlayStation 4.

Con el Black Friday a la vuelta de la esquina nos hemos vuelto un poco locos. A qué estás esperando, jentra en Wuolah.com y participa!



Smartband Fitbit Flex 2.

- La asignación de memoria principal la gestionan conjuntamente el SO y el hardware de gestión de memoria.
- El SO gestiona el uso de dispositivos E/S y el acceso a ficheros.
- **Habilidad de evolucionar.** Razones para hacerlo:
 - Actualizaciones del hardware actual y nuevos tipos de hardware.
 - Mejorar y/o aportar nuevos servicios.
 - Resolución de fallos de un SO a lo largo del tiempo de uso.

1. Introducción

1.1. Evolución de los SOs

Procesamiento serial

Con los primeros ordenadores (1940-1950) los programadores interactuaban directamente con el hardware mediante luces, interruptores, algún dispositivo de entrada (ej: lector de cartas) y una impresora. No había SO.

Problemas: planificación de recursos y un inmenso período de tiempo en organización y configuración.

Solución: inclusión de herramientas software como bibliotecas, depuradores y rutinas E/S.

Sistemas monoprogramados

A partir de mediados de los 50 se introducen SO simples con rutinas y subrutinas cargadas en memoria principal. Los primeros sistemas monoprogramados consistían en **cargar uno por uno los procesos en memoria**, es decir, se cargaba el primero y, cuando había acabado, o bien de forma normal o bien por error, se cargaba el siguiente proceso, y así sucesivamente.

Problemas: los resultados eran un caos, no se aprovechaba la CPU, el uso de memoria ni el uso de dispositivos de E/S, grandes tiempos de espera...

Sistemas multiprogramados

Los sistemas monoprogramados aún no eran suficiente, ya que eran muy lentos e ineficientes, ya que el procesador quedaba bloqueado o inactivo durante, por ejemplo, operaciones E/S.

Se introdujo entonces el concepto de multiprogramación, el cual consiste en que **varios procesos residen en memoria principal** y se ejecutan solapándose entre sí a tal velocidad que **se están ejecutando simultáneamente** (en realidad se ejecutan consecuentemente). Esto reducía los tiempos de espera enormemente ya que mientras un proceso esperaba a alguna operación E/S, otros procesos podían estar ejecutándose.

Este sistema, que era mucho más sofisticado, requería contar con un mecanismo de interrupciones y un gestor de memoria, además de algoritmos de selección de CPU.

Sistemas de tiempo compartido

Cuando los **procesos**, aparte de estar situados **en memoria al mismo tiempo** y estar ejecutándose consecuentemente, pueden ser o son **interactivos**, nos encontramos frente a la técnica que se conoce como tiempo compartido.

Esta técnica permite que el SO pueda expulsar al proceso actual una vez acabado su tiempo arbitrario (quantum) para dar paso a otros.

Permite que un gran número de usuarios interactúen y ejecuten diversos programas de manera simultánea. También necesita una serie de requisitos.

1.2. Características de un buen SO

- **Corrección.** Habilidad de corregir errores detectados durante la ejecución. Ejemplo:
 - Debe contener la sincronización adecuada requerida por los procesos en cuestión (programas concurrentes). ¿Qué eventos pueden provocar que varias partes del SO se ejecuten simultáneamente?
- **Interrupción:** es una señal recibida por el procesador para indicarle que debe interrumpir el curso actual de ejecución y pasar a ejecutar código específico para tratar esa situación. Tras la ejecución de dicho bloque de instrucciones se restablece la ejecución casual. Se tratan de señales asíncronas y existen dos tipos:
 - Interrupción hardware: son externas a la CPU y por lo general vienen de dispositivos E/S.

- **Interrupción software:** son generadas por un programa en ejecución. También reciben el nombre de llamadas al sistema.
- **Excepción:** se trata de una señal síncrona típicamente causada por un error en un programa. Genera un cambio de contexto (*usuario* → *supervisor*) para que el SO atienda el error. Se trata de mecanismos de protección.
- Distintas partes del SO pueden estar implementadas como procesos concurrentes.
 - Ser capaz de detectar todo tipo de errores.
 - Ser capaz de funcionar correctamente ante una caída del flujo eléctrico (conjunto de instrucciones que deben ejecutarse “Todo o nada”).
- **Tolerancia a fallos.** No todos los errores que surjan pueden ser tratados por igual ni generar las mismas consecuencias para el sistema.
- **Extensible.** Debido a los nuevos y rápidos cambios que se producen en hardware y software gracias a los avances tecnológicos, un SO debe poder incluir, añadir, modificar o eliminar componentes a un coste mínimo y de forma sencilla.
- **Reentrancia.** Capacidad de que pueda haber varias instancias de ejecución.
- **Fiabilidad.** El SO debe ser robusto, es decir, debe responder de forma predecible a circunstancias de error y protegerse activamente a sí mismo y los usuarios de acciones accidentales y malintencionadas.
- **Eficiencia.**

1.3. Servicios / Componentes del SO

Gestor de procesos

Un proceso es una instancia de un programa en ejecución con un estado actual y una serie de recursos asignados. El SO es responsable, entre otros, de las actividades relacionadas a procesos:

- Creación / Destrucción
- Suspender / Reanudar
- Sincronización / Comunicación
- Controlar hebras

Gestor de memoria principal

La memoria principal es el almacenamiento de acceso directo para los procesos y los dispositivos de E/S. Esta memoria es volátil. Es responsable, entre otras, de:

- Asignar / desasignar memoria a los programas
- Mantener la pista de la memoria utilizada actualmente y quién la usa
- Decidir cuánta memoria asignar a cada proceso y cuándo debe ser retirado de memoria
- Transformación direcciones virtuales <—> direcciones reales
- Sistemas de paginación / segmentación

Gestor de archivos

Un archivo es una colección de información con nombre y la entidad básica de almacenamiento persistente. El sistema de almacenamiento suministra:

- Primitivas para manipular archivos y directorios, crear, borrar, leer, escribir...
- Correspondencia entre archivos y un almacenamiento secundario
- Otros servicios generales (ej: backup)

Gestor de E/S

Los SOs ofrecen a los programas un interfaz estándar de los dispositivos, es decir, utilizan las mismas funciones independientemente del dispositivo al que acceden.

Un controlador de dispositivo gestiona un tipo de dispositivo: inicialización, interrupciones, lectura/escritura...

Sistema de protección

Referencia al mecanismo para controlar los accesos de los programas a los recursos del sistema. Es un mecanismo general a todo el SO, es decir, no está localizado en un único módulo.

Éste mecanismo debe:

- Distinguir entre uso autorizado y no autorizado
- Especificar qué control se debe suponer
- Suministrar los medios para su aplicación

Gestor de recursos

Debe repartir los recursos entre los procesos según estos tres principios:

- Equidad
- Grado de importancia y requisitos
- Eficiencia

2. Sistemas de propósito específico

SO de tiempo real

Suelen utilizarse en aplicaciones especializadas y si el objetivo es **garantizar que las respuestas se produzcan en un período de tiempo (t) especificado** (deadlines). Hay dos tipos:

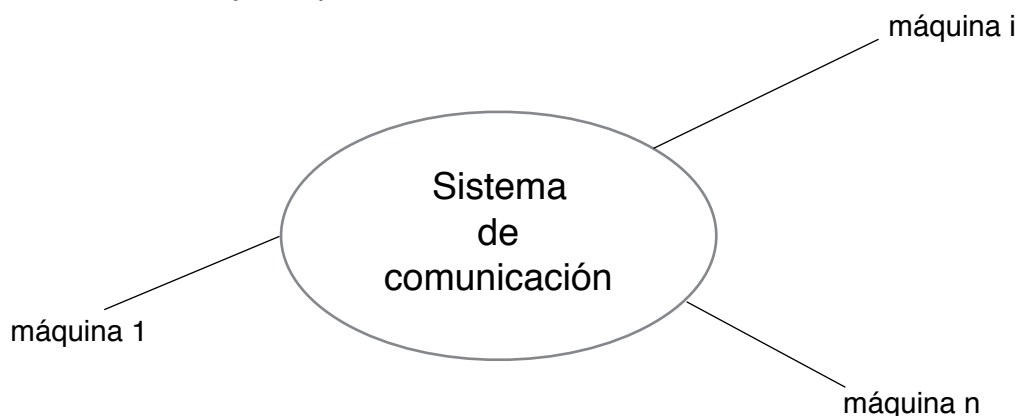
- **SO de tiempo real estricto:** es absolutamente imperativo que las respuestas se produzcan en el tiempo especificado.
- **SO de tiempo real no estricto:** no es totalmente necesario que las respuestas se produzcan en los plazos.

Con el uso de aplicaciones multimedia sobre PCs, todos los SOs actuales tienen requisitos en tiempo real.

Problema: un SO de este tipo debe poder ejecutar (planificar) todas las actividades del sistema con el fin de satisfacer todos los requisitos críticos de las mismas, incluidos los temporales.

Sistema informático distribuido

Es un sistema multicomputador que **no posee una memoria ni un reloj común**, cuyo objetivo básico es compartir recursos distribuidos tanto hardware como software. Como poseen una memoria y tiempo no comunes, necesita una contracción diferente.



Hace transparente a los usuarios dónde se ubican los archivos y dónde se ejecutan los procesos.

Por último, permiten un aumento de la fiabilidad del sistema. Si una parte falla, el resto puede seguir su ejecución, al menos parcialmente.

SO en red

Es un sistema operativo clásico que permite la **interconexión de ordenadores** para poder acceder a servicios y recursos hardware y software, creando redes de computadores. Posibilita la comunicación con otros equipos en el ámbito de una red (internet o intranet).

SO paralelo

Exclusivos para sistemas multiprocesadores, es decir, sistemas de computador con **varios procesadores que comparten una única memoria y reloj**.

Estos sistemas sustentan aplicaciones paralelas cuyo objetivo es aumentar la velocidad de tareas computacionalmente complejas (mejorar la eficiencia).

Actualmente implementa **multiprocesamiento simétrico (SMP)** que consiste en que todos los procesadores pueden ejecutar tanto código del SO como de las aplicaciones, sin importar la localización en memoria. Estos sistemas pueden mover fácilmente tareas entre los procesadores para garantizar eficientemente el trabajo.

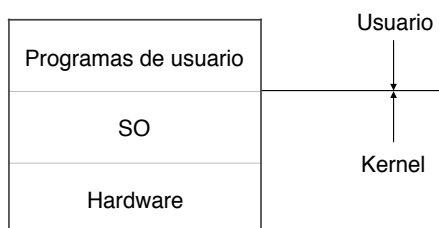
Problema: Para comunicarse con memoria, los procesadores comparten un bus, que debe ser arbitrado para que solo un procesador lo use en cada instante de tiempo. Esto conlleva largas esperas a que lleguen los datos de memoria.

3. Estructura / Arquitectura de un SO

Cuando hablamos de estructura nos referimos a la estructura con la cual construimos el software (componentes del mismo y como se relacionan). Al diseñar la estructura de un sistema, hay que hacerlo cuidadosamente para que funcione apropiadamente y pueda modificarse con facilidad.

Vamos a ver varios tipos:

3.1. Estructura monolítica



Este sistema operativo se describe como una colección de procedimientos enlazados entre sí en un único archivo ejecutable binario extenso (cientos de MB).

No tiene estructura interna y todo el código comparte el mismo lugar en memoria. Cualquier parte del mismo puede llamar a otra y por tanto la relación entre sus componentes es compleja.

Problemas

- Son **poco manejables y difíciles de comprender**, siendo, por tanto, **difíciles de modificar y mantener**.
- **No son para nada fiables**, un fallo de algún módulo puede provocar que caiga todo el sistema.
- **No hay ocultación de información**, ya que todos los módulos pueden llamar a cualquier otro.

Solución

- Tratar de **buscar otra estructura** que **simplifique el diseño** del sistema, así como la construcción, depuración, ampliación y mejora de funciones.

3.2. Estructura de capas

Programas de usuario
Capa n
...
Capa 1
Hardware

El sistema operativo se dispone como una serie de capas constituidas una encima de otra de forma jerárquica. Cada una es una máquina más abstracta para la capa superior, con lo que permite una ocultación "capa a capa".

Por modularidad, las capas se seleccionan para que cada una utilice solo funciones de las capas inferiores.

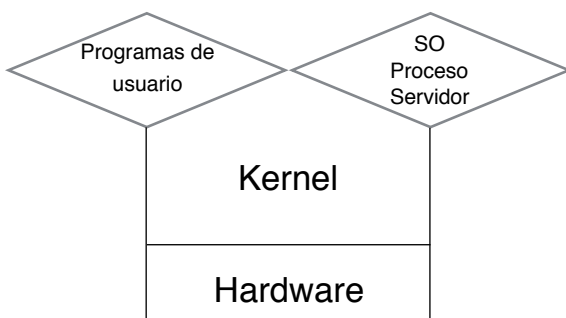
A la hora de depuración y construcción, esta estructura presenta una clara ventaja, ya que cada capa se construye y se depura y, mientras que haya errores, no se empieza con la capa siguiente.

Problemas

- Requiere una **planificación muy cuidadosa**, ya que es muy difícil organizar correctamente las capas.

- A menudo, los sistemas se modelan con esta estructura pero se construyen de otra manera.
- Sobrecarga de comunicaciones entre las capas.
- Menor eficiencia que otras implementaciones debido a que pasar de una capa a otra supone una carga de trabajo adicional.

3.3. Estructura microkernel / Cliente - Servidor



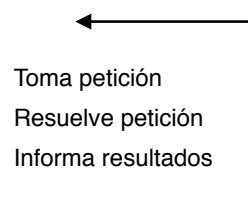
Núcleo. Software que:

- Tiende a ser residente en memoria.
- Tiende a ejecutarse en máximo nivel de privilegio (modo supervisor).
- Tiende a ejecutarse en alto nivel de inhibición de interrupciones.

Para implementar un microkernel, se estructura el SO eliminando todos los componentes no esenciales del kernel y se implementan como programas de usuario.

El microkernel sólo debe contener información básica para **crear y comunicar procesos**. Esto último es clave. Debe proporcionar un mecanismo de comunicación entre procesos cliente y procesos servidor. La aplicación envía el mensaje de una solicitud (send) y permanece esperando la respuesta (receive). El kernel verifica este mensaje y lo manda al servidor. Por último, el servidor que está en espera realiza el servicio solicitado y devuelve el resultado.

Un proceso servidor es aquel que resuelve una cola de peticiones:



Las ventajas que ofrece esta estructura son:

- Es muy **extensible** y **personalizable**, es decir, podemos cambiar un servicio del SO cambiando el proceso que lo implementa.
- Mayor **fiabilidad** y **seguridad**, ya que un posible error de un servicio del SO queda confinado en el espacio de direcciones del proceso que lo implementa.

Problemas

- **Peor rendimiento.**
- Carga de procesamiento adicional por funciones del sistema.

Microkernel vs Monolítica

- La estructura microkernel es **más flexible** que la monolítica.
- La estructura microkernel tiene **peor rendimiento** que la monolítica:

	Monolítica	Microkernel
Nº de cambios de nodo	2	4
Nº de cambios proceso - SO	2	4
Nº de cambios proceso - proceso	0	2