

Sistemas Operativos

Miguel Angel Robles Urquiza
Antonio Martín Ruiz

2016-2017

1 Estructura o Arquitectura del sistema

- Estructura del sistema

El software debe ser modular. Esto ayudará a organizar el proceso de desarrollo de software y limitará el esfuerzo de diagnosticar y corregir errores. Los módulos deben tener interfaces bien definidas, y estas interfaces deben ser tan sencillas como sea posible.

Existen sistemas operativos sin una estructura definida. Son sistemas simples y limitados. Algunos ejemplo son MS-DOS y el UNIX original. Estas estructuras se desarrollan hacia organizaciones jerárquicas al crecer las funcionalidades del sistema operativo, por lo que en la actualidad son obsoletas.

La estructura jerárquica de un sistema operativo moderno separa sus funciones e acuerdo a las características de su escala de tiempo y su nivel de abstracción. Se puede ver el sistema como una serie de niveles. Cada nivel realiza un subconjunto relacionado de funciones requeridas por el sistema operativo. Dicho nivel confía en los niveles inmediatamente inferiores para realizar funciones más primitivas y ocultar los detalles de esas funciones. Cada nivel proporciona servicios a la capa inmediatamente superior. Idealmente, estos niveles deben definirse de tal forma que los cambios en un nivel no requieran cambios en otros niveles.

Modelo de sistema operativo jerárquico:

Nivel	Nombre	Objetos
13	Intérprete de mandos	Entorno de programación de usuario
12	Proceso de usuario	Procesos de usuario
11	Directorios	Directorios
10	Dispositivos	Pantallas, impresoras...
9	Sistemas de ficheros	Ficheros
8	Comunicadores	Tuberías
7	Memoria Virtual	Segmentos, páginas
6	Almacenamiento secundario local	Bloques de datos y canales
5	Procesos primitivos	Procesos primitivos, semáforos, lista...
4	Interrupciones	Programas de gestión de interrupciones
3	Procedimientos	Pila de llamadas, registro de activación
2	Conjunto de instrucciones	Pila de evaluación, intérprete...
1	Circuitos electrónicos	Registros, puertas, buses

- Categorías de SO

- Arquitectura micrókernel: asigna sólo unas pocas funciones esenciales al núcleo, incluyendo los espacios de almacenamiento, comunicación entre procesos (IPC), y la planificación básica. Ciertos procesos proporcionan otros servicios del sistema operativo, algunas veces denominados servidores, que ejecutan en modo usuario y son tratados como cualquier otra aplicación por el micrókernel. Esta técnica desacopla el núcleo y el desarrollo del servidor. En esencia, un micrókernel interactúa con procesos locales y remotos del servidor de la misma forma, facilitando la construcción de los sistemas distribuidos.
- Multithreading: técnica en la cual un proceso, ejecutando una aplicación, se divide en una serie de hilos o threads que pueden ejecutar concurrentemente.
 1. Thread o hilo: Se trata de una unidad de trabajo. Incluye el contexto del procesador (PC y puntero de pila) y su propia área de datos para una pila. Un hilo se ejecuta secuencialmente y se puede interrumpir de forma que el procesador pueda dar paso a otro hilo.
 2. Proceso: Es una colección de uno o más hilos y sus recursos de sistema asociados. Dividiendo una sola aplicación en múltiples hilos, el programador tiene gran control sobre la modularidad de las aplicaciones y la temporización de los eventos relacionados con la aplicación. Es útil para las aplicaciones que llevan a cabo un número de tareas esencialmente independientes.

- Multiprocesamiento simétrico (SMP: Symmetric MultiProcessing): es un sistema de computación aislado con las siguientes características:
 1. Tiene múltiples procesadores.
 2. Estos procesadores comparten las mismas utilidades de memoria principal y de E/S, interconectadas por un bus de comunicación u otro esquema de conexión interna.
 3. Todos los procesadores pueden realizar las mismas funciones.

Ventajas:

- * Rendimiento. Con la multiprogramación, sólo un proceso puede ejecutar a la vez; mientras tanto, el resto de los procesos esperan por el procesador. Con multiproceso, más de un proceso puede ejecutarse simultáneamente, cada uno de ellos en un procesador diferente.
- * Disponibilidad. En un multiprocesador simétrico, debido a que todos los procesadores pueden llevar a cabo las mismas funciones, el fallo de un solo procesador no para la máquina, pero puede continuar funcionando con un rendimiento reducido.
- * Crecimiento incremental. Un usuario puede mejorar el rendimiento de un sistema añadiendo un procesador adicional.
- * Escalado. Los fabricantes pueden ofrecer un rango de productos con diferente precio y características basadas en el número de procesadores configurado en el sistema.
- SO distribuido: proporciona la ilusión de un solo espacio de memoria principal y un solo espacio de memoria secundario. Otra innovación en el diseño de los sistemas operativos es el uso de tecnologías orientadas a objetos.
- Diseño orientado a objetos: introduce una disciplina al proceso de añadir extensiones modulares a un pequeño núcleo. Permite a los programadores personalizar un sistema operativo sin eliminar la integridad del sistema. Facilita el desarrollo de herramientas distribuidas y sistemas operativos distribuidos.

- **Micronúcleos**

Un **micronúcleo** es la pequeña parte central de un sistema operativo que proporciona las bases para extensiones modulares.

La filosofía existente en el micronúcleo es que solamente las funciones absolutamente esenciales del sistema operativo estén en el núcleo. Los servicios y aplicaciones menos esenciales se construyen sobre el micronúcleo y se ejecutan en modo usuario. Muchos servicios que tradicionalmente habían

formado parte del sistema operativo ahora son subsistemas externos que interactúan con el núcleo y entre ellos mismos.

La arquitectura del micronúcleo reemplaza la tradicional estructura vertical y estratificada en capas por una horizontal. Los componentes del sistema operativo externos al micronúcleo se implementan como servidores de procesos; interactúan entre ellos dos a dos, normalmente por paso de mensajes a través del micronúcleo. De esta forma, el micronúcleo funciona como un intercambiador de mensajes. El micronúcleo previene el paso de mensajes a no ser que el intercambio esté permitido.

Ventajas:

1. **Interfaz uniforme**, en las peticiones realizadas por un proceso. Los procesos no necesitan diferenciar entre servicios a nivel de núcleo y a nivel de usuario.
2. **Extensibilidad**, permitiendo agregar nuevos servicios, así como la realización de múltiples servicios en la misma área funcional. Con la arquitectura micronúcleo, cuando se añade una nueva característica, sólo el servidor relacionado necesita modificarse o añadirse.
3. **Flexibilidad**, no sólo se pueden añadir nuevas características al sistema operativo, además las características existentes se pueden eliminar para realizar una implementación más pequeña y más eficiente.
4. **Portabilidad**, en la arquitectura micronúcleo, todo o gran parte del código específico del procesador está en el micronúcleo. Por tanto, los cambios necesarios para transferir el sistema a un nuevo procesador son menores y tienden a estar unidos en grupos lógicos.
5. **Fiabilidad**, el uso de un pequeño número de interfaces de programación de aplicaciones (API) hace más sencillo producir un código de calidad fuera del núcleo. El programador tiene un número limitado de API que dominar y métodos limitados de interacción. Es más difícil afectar negativamente a otros componentes del sistema.
6. **Soporte de sistemas distribuidos** El micronúcleo nos lleva por sí mismo al soporte de sistemas distribuidos, incluyendo clusters controlados por sistemas operativos distribuidos. Un proceso puede enviar un mensaje sin saber en qué máquina reside el servicio pedido.
7. **Sistema operativo orientado a objetos**, una arquitectura micronúcleo funciona bien en el contexto de un sistema operativo orientado a objetos.

Una potencial **desventaja** que se cita a menudo de los micronúcleos es la del *rendimiento*. Lleva más tiempo construir y enviar un mensaje a través del micronúcleo, y aceptar y decodificar la respuesta, que hacer una simple llamada a un servicio.

Con respecto al **diseño**, el micronúcleo debe incluir aquellas funciones que dependen directamente del hardware y aquellas funciones necesarias para mantener a los servidores y aplicaciones operando en modo usuario.

La forma básica de **comunicación** entre dos procesos o hilos en un sistema operativo con micronúcleo son los mensajes. Un mensaje incluye una cabecera que identifica a los procesos remitente y receptor y un cuerpo que contiene directamente los datos, un puntero a un bloque de datos, o alguna información de control del proceso.

Con una arquitectura micronúcleo es posible manejar las **interrupciones** hardware como mensajes e incluir los puertos de E/S en los espacios de direcciones. El micronúcleo puede reconocer las interrupciones pero no las puede manejar.

- **Computación en tiempo real**

La computación de tiempo real puede definirse como aquella en la que la corrección del sistema depende no sólo del resultado lógico de la computación sino también del momento en el que se producen los resultados. Podemos definir un sistema de tiempo real definiendo lo que se entiende por proceso de tiempo real, o tarea. En general, en un sistema de tiempo real, algunas de las tareas son tareas de tiempo real, y éstas tienen cierto grado de urgencia. Tales tareas intentan controlar o reaccionar a eventos que tienen lugar en el mundo exterior.

- Una **tarea de tiempo real duro** es aquella que debe cumplir su plazo límite.
- Una **tarea de tiempo real suave** tiene asociado un plazo límite deseable pero no obligatorio
- Una *tarea aperiódica* tiene un plazo en el cual debe finalizar o comenzar, o puede tener una restricción tanto de su instante de comienzo como de finalización.
- Una *tarea periódica* en la que el requisito puede ser enunciado como «una vez por periodo T» o «exactamente T unidades a parte».

Los sistemas operativos de tiempo real pueden ser caracterizados por tener requisitos únicos en cinco áreas generales:

1. **Determinismo:** realiza operaciones en instantes de tiempo fijos predeterminados o dentro de intervalos de tiempo predeterminados y si no se cumple se producen daños al sistema.
2. **Reactividad:** se preocupa de cuánto tiempo tarda el sistema operativo, después del reconocimiento, en servir la interrupción.
3. **Control del usuario:** es generalmente mucho mayor en un sistema operativo de tiempo real que en sistemas operativos ordinarios.

4. **Fiabilidad:** es normalmente mucho más importante para los sistemas de tiempo real que para los que no lo son.
5. **Operación de fallo suave:** es una característica que se refiere a la habilidad del sistema de fallar de tal manera que se preserve tanta capacidad y datos como sea posible.
6. Eficiente, comodo y evolutivo

Para cumplir los requisitos precedentes, los sistemas operativos de tiempo real incluyen de forma representativa las siguientes características]:

- Cambio de proceso o hilo rápido.
- Pequeño tamaño (que está asociado con funcionalidades mínimas).
- Capacidad para responder rápidamente a interrupciones externas.
- Multitarea con herramientas para la comunicación entre procesos como semáforos, señales y eventos.
- Utilización de ficheros secuenciales especiales que pueden acumular datos a alta velocidad.
- Planificación expulsiva basada en prioridades.
- Minimización de los intervalos durante los cuales se deshabilitan las interrupciones.
- Primitivas para retardar tareas durante una cantidad dada de tiempo y para parar/retomar tareas.
- Alarmas y temporizaciones especiales.

Multihilo se refiere a la capacidad de un sistema operativo de dar soporte a múltiples hilos de ejecución en un solo proceso. Dentro de un proceso puede haber uno o más hilos, cada uno con:

- Un estado de ejecución por hilo.
- Un contexto de hilo que se almacena cuando no está en ejecución. es como un contador de programa independiente dentro de un proceso.
- Una pila de ejecución.
- Acceso a la memoria.

En un entorno multihilo, sigue habiendo un único bloque de control del proceso y un espacio de direcciones de usuario asociado al proceso, pero ahora hay varias pilas separadas para cada hilo. Los mayores beneficios de los hilos provienen de las consecuencias del rendimiento:

1. Lleva mucho menos tiempo crear un nuevo hilo en un proceso existente que crear un proceso totalmente nuevo.
2. Lleva menos tiempo finalizar un hilo que un proceso.

3. Lleva menos tiempo cambiar entre dos hilos dentro del mismo proceso.
4. Los hilos mejoran la eficiencia de la comunicación entre diferentes programas que están ejecutando.

A veces los hilos son también útiles en un solo procesador ya que ayudan a simplificar la estructura de programas que realizan varias funciones diferentes.

Sistemas operativos distribuidos y de red

Cada vez es más extendido el uso de los computadores personales y su asociación con los servidores. Los computadores personales se utilizan para dar soporte a aplicaciones de fácil manejo, tales como procesadores de texto, hojas de cálculo y presentación de gráficos. Los servidores mantienen las bases de datos corporativas y sofisticados gestores de bases de datos y software de sistemas de información. De esta manera se llega al uso de los ordenadores personales como terminales de los servidores para una mayor integración de las utilidades de ambos dispositivos. Existen diferentes soluciones para esta necesidad:

- Arquitectura de comunicaciones. Es un software que da soporte a un grupo de computadores en red. Proporciona soporte para aplicaciones distribuidas. Sin embargo, los computadores siguen siendo entidades independientes para los usuarios y para las aplicaciones, que se deben comunicar entre sí por expreso deseo. La arquitectura de comunicaciones más ampliamente utilizada es el conjunto de protocolos TCP/IP.
- Sistema operativo de red. En esta configuración hay una red de máquinas, normalmente estaciones de trabajo de un solo usuario, y una o más máquinas servidoras. Éstas proporcionan servicios de red o aplicaciones. Cada computador tiene su propio sistema operativo. El sistema operativo de red es un añadido al sistema operativo local, que permite a las máquinas interactuar con los servidores.
- Sistema operativo distribuido. Es un sistema operativo común compartido por una red de computadores. A los usuarios les parece un sistema operativo normal centralizado, pero les proporciona acceso transparente a los recursos de diversas máquinas.

El intercambio de información entre computadores con finalidad cooperativa se conoce como comunicación de computadores. De forma similar, cuando uno o más computadores se interconectan a través de una red de comunicación, el conjunto de computadores se denomina red de computadores. Ya que se requiere un nivel similar de cooperación entre un terminal y un computador, estos términos también se utilizan cuando alguna de las entidades de comunicación son terminales. En relación a la comunicación de computadores y redes de computadores, hay dos conceptos de suma

importancia: protocolos y arquitectura de comunicaciones o arquitectura de protocolos.

Un **protocolo** se utiliza para comunicar entidades de diferentes sistemas. En general, una entidad es cualquier cosa capaz de enviar y recibir información, y un sistema es un objeto físico que contiene una o más entidades. Lo que se comunica, cómo se comunica y cuándo se comunica, debe hacerse de acuerdo a unas convenciones entre las entidades involucradas. Las convenciones se denominan protocolos, que se pueden definir como un conjunto de reglas que gobiernan el intercambio de datos entre dos entidades. Los elementos principales de un protocolo son los siguientes:

- Sintaxis. Incluye cosas tales como formatos de datos y niveles de señales.
- Semántica. Incluye información de control para realizar coordinación y gestión de errores.
- Temporización. Incluye ajuste de velocidades y secuenciamiento.

Conociendo el concepto de protocolo, se puede introducir el concepto de **arquitectura de protocolos**. Consiste en dividir la tarea de comunicación de computadores en diferentes módulos que se encargan de una parte específica. El módulo de transferencia de ficheros contiene toda la lógica que es única a la aplicación de transferencia de ficheros, tal como la transmisión de claves, mandatos y registros de los ficheros. El módulo de servicios de comunicaciones se preocupa de que los dos sistemas de computación estén activos y listos para el intercambio de datos y de llevar la cuenta de los datos que se están intercambiando para asegurar su entrega. La lógica para la comunicación con la red se encuentra en el módulo de acceso a red.

Sistemas operativos multiprocesador

Tradicionalmente el computador ha sido visto como una máquina secuencial. Los algoritmos descritos por los lenguajes de programación son secuenciales, recorriendo el ordenador las órdenes de una en una. Sin embargo esta visión no es del todo correcta, ya que existen ejemplos de funciones realizadas en paralelo. Conforme la tecnología ha avanzado y ha disminuido el precio del software se ha optado por una mayor implementación del paralelismo para mejorar la fiabilidad y el rendimiento. Una de los enfoques más populares es el multiprocesamiento simétrico (SMP).

Un multiprocesador simétrico consiste en un conjunto de procesadores que comparten memoria. El kernel del sistema operativo puede ejecutarse en cualquiera de los procesadores. EL núcleo puede construirse como varios procesos o hilos, permitiendo la ejecución de diferentes partes del núcleo en paralelo. Esto conlleva el uso de técnicas para sincronizar el funcionamiento de los procesadores.

Un sistema operativo SMP gestiona los procesadores y otros recursos del computador, de manera que el usuario puede ver al sistema de la misma forma que si fuera un sistema uniprocador multiprogramado. Un usuario puede desarrollar aplicaciones que utilicen múltiples procesos o múltiples hilos dentro de procesos sin preocuparse de si estará disponible un único procesador o múltiples procesadores. De esta forma, un sistema operativo multiprocador debe proporcionar toda la funcionalidad de un sistema multiprogramado, además de características adicionales para adecuarse a múltiples procesadores. Las principales claves de diseño incluyen las siguientes características:

- Procesos o hilos simultáneos concurrentes. Las rutinas del núcleo necesitan ser reentrantes para permitir que varios procesadores ejecuten el mismo código del núcleo simultáneamente.
- Planificación. La planificación se puede realizar por cualquier procesador, por lo que se deben evitar los conflictos. Si se utiliza multihilo a nivel de núcleo, existe la posibilidad de planificar múltiples hilos del mismo proceso simultáneamente en múltiples procesadores.
- La sincronización es un servicio que fuerza la exclusión mutua y el orden de los eventos. Un mecanismo común de sincronización que se utiliza en los sistemas operativos multiprocador son los cerrojos.
- Gestión de memoria. el sistema operativo necesita explotar el paralelismo hardware existente, como las memorias multipuerto, para lograr el mejor rendimiento. Los mecanismos de paginación de los diferentes procesadores deben estar coordinados para asegurar la consistencia cuando varios procesadores comparten una página o segmento y para decidir sobre el reemplazo de una página.
- Fiabilidad y tolerancia a fallos. El sistema operativo no se debe degradar en caso de fallo de un procesador. El planificador y otras partes del sistema operativo deben darse cuenta de la pérdida de un procesador y reestructurar las tablas de gestión apropiadamente.

Funcionalidad de los hilos

Hay cuatro operaciones básicas relacionadas con los hilos que están asociadas con un **cambio de estado** del hilo:

- Creación. Cuando se crea un nuevo proceso, también se crea un hilo de dicho proceso. Posteriormente, un hilo del proceso puede crear otro hilo dentro del mismo proceso, proporcionando un puntero a las instrucciones y los argumentos para el nuevo hilo. Al nuevo hilo se le proporciona su propio registro de contexto y espacio de pila y se coloca en la cola de Listos.
- Bloqueo. Cuando un hilo necesita esperar por un evento se bloquea, almacenando los registros de usuario, contador de programa y punteros de pila. El procesador puede pasar a ejecutar otro hilo en estado Listo, dentro del mismo proceso o en otro diferente.

- Desbloqueo. Cuando sucede el evento por el que el hilo está bloqueado, el hilo se pasa a la cola de Listos.
- Finalización. Cuando se completa un hilo, se liberan su registro de contexto y pilas.