TEMA 1. ESTRUCTURA DE LOS SO

Sistema monolítico

Los **sistemas monolíticos** son aquellos en los que su centro es un grupo de estructuras fijas, las cuales funcionan entre sí, para poder tener esta estructura, las diferentes partes del kernel son compiladas por capas.

Los sistemas monolíticos se describen en **3 procesos principales**:

- Un Programa principal que invoca el procedimiento de servicio solicitado.
- Un Conjunto de procedimientos de servicio que llevan a cabo las llamadas del sistema.
- Un Conjunto de procedimientos de utilería que ayudan a los procedimientos del servicio.

Los núcleos monolíticos proporcionan la mayor parte de las funcionalidades propias del sistema operativo, incluyendo la planificación, los sistemas de ficheros, las redes, los controladores de dispositivos, la gestión de memoria y otras funciones. El SO está formado por un conjunto de procedimientos de forma que cada uno puede llamar a los demás cuando lo necesite. Estos procedimientos se ejecutan en modo supervisor. Normalmente, un núcleo monolítico se implementa como un único proceso con todos los elementos compartiendo el mismo espacio de direcciones.

El **problema** de los núcleos monolíticos es que son difíciles de comprender, modificar y mantener. No son fiables: un error en alguna parte puede provocar la caída del sistema.

Se podría implementar como un **sistema por capas**, donde cada capa es una máquina más abstracta para la capa superior. Por modularidad, las capas se seleccionan para que cada una utilice funciones sólo de las capas inferiores.

Ejemplo de sistema por capas: **El Sistema THE**

El sistema estaba compuesto de una serie de procesos secuenciales.

Nivel 5	Programas de usuario
Nivel 4	Búfering para dispositivos de E/S
Nivel 3	Manejador de consola del operador
Nivel 2	Gestión de memoria
Nivel 1	Planificación de la CPU
Nivel 0	Hardware

Problemas de THE:

- Los sistemas de capas deben ser jerárquicos pero los sistemas reales son

más complejos. Por ejemplo: el sistema de archivos podría ser un proceso en la capa de memoria virtual y la capa de memoria virtual podría usar archivos como almacén de apoyo de E/S.
- Sobrecarga de comunicaciones entre procesos de distintas capas
- A menudo, los sistemas están modelados con esta estructura pero no están así construidos

Microkernel

Una **arquitectura micronúcleo** 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 micronúcleo

El sistema operativo se reduce a un núcleo mínimo, se implementan la mayoría de las funciones del SO como procesos de usuario, lo que proporciona mayor flexibilidad. Para solicitar un servicio, el proceso de usuario (cliente) envía un mensaje al proceso servidor, que realiza el servicio y devuelve al cliente una respuesta, provocando una mayor sobrecarga por en envío/recepción de mensajes.

Beneficios:

- Su arquitectura facilita la extensibilidad, permitiendo agregar nuevos servicios en la misma área funcional.
- No solo se pueden añadir nuevas características al SO, además las características existentes se pueden eliminar para realizar una implementación más pequeña y eficiente.
- Todo o gran parte del código específico del procesador está en el microkernel.

Una de las **desventajas** más conocida del microkernel es la del rendimiento. Esto depende del tamaño y la funcionalidad del microkernel.

Multithreading

Multithreading es una técnica en la cual un proceso, ejecutando una aplicación, se divide en una serie de hilos o threads que pueden ejecutar concurrentemente.

- **Thread o hilo**. Se trata de una unidad de trabajo. Incluye el contexto del procesador (que contiene el contador del programa y el puntero de pila) y su propia área de datos para una pila (para posibilitar el salto a subrutinas). Un hilo se ejecuta secuencialmente y se puede interrumpir de forma que el procesador pueda dar paso a otro hilo.
- **Proceso**. Es una colección de uno o más hilos y sus recursos de sistema asociados (memoria, conteniendo tanto código, como datos, ficheros abiertos

y dispositivos). Esto corresponde al concepto de programa en ejecución. La técnica multithreading es útil para las aplicaciones que llevan a cabo un número de tareas independientes.

Máquinas virtuales

Una **máquina virtual** es un software que implementa una máquina virtual (igual o distina a la máquina real). Se trata de abstraer el hardware de la computadora formando entornos de ejecución diferentes, creando la ilusión de que cada entorno de ejecución está en una computadora diferente. La máquina virtual no proporciona funcionalidades adicionales sino que proporciona una interfaz "idéntica" al hardware básico. Una petición de servicio es atendida por la copia de SO sobre la que se ejecuta. La capacidad de procesamiento actual mitiga la ineficiencia de las Máquinas Virtuales. Los procesadores más actuales incluyen soporte para la misma.

Beneficios:

- Más seguras, no es posible la compartición directa de recursos.
- Investigación y desarrollo de sistemas operativos.

Sistemas Operativos de propósito específico

Sistemas Operativos de 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.

En un **sistema de tiempo real** se producen unas tareas que dan soluciones a problemas que vienen del mundo exterior y que se producen en un espacio de tiempo determinado, pudiéndose establecer así un margen de tiempo límite para realizar las tareas. De este modo se pueden dividir las tareas en tareas de **tiempo real duro** si tienen que realizarse en un tiempo determinado y un exceso de tiempo en dicha actividad puede suponer un error grave para el sistema, o tareas de **tiempo real blando** cuyo plazo de tiempo no es estricto y la tarea se puede seguir ejecutando después de haber cumplido el plazo sin desencadenar errores graves.

Estas tareas se distinguen a su vez dada la periodicidad de su ejecución, de este modo hay **tareas aperiódicas** si su ejecución es única y está restringida a un plazo de tiempo, o **tareas periódicas** si se repiten n veces en un espacio de tiempo o una vez cada período t.

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

- Determinismo
- Reactividad
- Control del usuario
- Fiabilidad
- Operación de fallo suave

El **determinismo** (tiempo de SO en reconocer la interrupción) nos dice que unas tareas concretas van a tardar un tiempo concreto en ser finalizadas, un sistema no puede ser completamente determinista, pues se ve limitado por sus recursos y por la prioridad de las tareas a ejecutar, se centra en el tiempo previo al reconocimiento previo de una instrucción.

La **reactividad**, relacionada con el determinismo, se enfoca en el tiempo después del reconocimiento que tarda el SO en servir a esa interrupción, además incluye algunos aspectos como:

1. La cantidad de tiempo necesario para ejecutar la rutina de servicio de la interrupción actual, RSI (si es necesario cambiar de contexto requerirá un tiempo adicional).
2. El tiempo que requiere para procesarla, que se define a nivel hardware.
3. El efecto anidamiento. Si una interrupción puede ser interrumpida por otra.

El **control del usuario** es generalmente mucho mayor en un sistema operativo de tiempo real que en sistemas operativos ordinarios. El usuario tiene un control sobre la prioridad de la tarea. El usuario distingue las tareas y se encarga de gestionar algunas características de gestión de memoria.

La **fiabilidad** en sistemas de tiempo real es mucho más trascendental que en los sistemas que no son de tiempo real, pues en estos se puede solucionar el problema re-arrancando el sistema, en los de tiempo real la degradación de los servicios ofrecidos a un sistema que responde a tareas en tiempo real puede ser mucho más catastrófica.

La **operación de fallo suave** es una característica de los sistemas para responder ante un fallo y ser capaz de minimizar las pérdidas sufridas, manteniendo la consistencia de ficheros y datos cuando sea posible. Además un sistema se considera estable cuando a pesar de no ser capaz de responder a todas las peticiones, es capaz de responder a las tareas más críticas.

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).
- Respuesta rápida 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.

En la mayoría de los casos los sistemas no son capaces de lidiar con todas las tareas asignadas, por lo que una de las principales áreas de trabajo hoy en día son los planificadores de tiempo real. Lo importante es que las tareas de tiempo real duro se completen y finalizar el máximo número de tareas de tiempo real blando.

Estructuras distribuidas

Hay una tendencia al proceso de datos distribuidos. Los procesadores, datos y procesamiento de datos pueden estar diseminados por toda la organización. Implica dividir funciones y organizar las bases de datos control de dispositivos,... Los computadores dependen de su asociación con los servidores. Existen distintas estructuras distribuidas: arquitectura multicomputador, sistemas operativos de red y sistemas operativos distribuidos.

Arquitectura Multicomputador

La **arquitectura de comunicaciones** es un software que da soporte a un grupo de computadores independientes, en red. Proporciona soporte para aplicaciones distribuidas, tales como correo electrónico, transferencia de ficheros y acceso a terminales remotos. Cada cual puede tener su propio SO y sólo se pueden comunicar directamente por deseo expreso.

Sistemas Operativos de Red

Un **sistema operativo de red** normalmente lo componen un único usuario con una o varias máquinas de servidores, que proporcionan acceso a servicios y aplicaciones, el usuario conoce la existencia de múltiples computadores y debe trabajar con ellos de forma explícita. Habitualmente se utiliza una arquitectura de comunicaciones común para dar soporte a estas aplicaciones de red. Lo que les diferencia de los SO de un solo procesador es la necesidad de software especial como: controlador de interfaz de la red; programas de conexión y acceso a archivos remotos.

Sistemas operativos Distribuidos

Un **sistema operativo distribuido** es un sistema operativo común compartido por una red de computadores. A los usuarios les proporciona acceso transparente a los recursos de diversas máquinas. Un sistema operativo distribuido puede depender de una arquitectura de comunicaciones para las funciones básicas de comunicación, pero normalmente se incorporan un conjunto de funciones de comunicación más sencillas para proporcionar mayor eficiencia.

Para hacer un intercambio de información entre dos computadores, ya sea algo tan sencillo como un fichero, es necesario establecer un enlace, tanto directo como en red de comunicaciones, pero además se necesitan tareas como:

1. Que el emisor active el enlace directo o informar a la propia red de comunicaciones
2. Debe comprobar que el receptor puede recibir estos datos
3. Que existe un programa que sea capaz de recepcionar el fichero
4. Que se encargue de realizar una traducción si las representaciones de datos son incompatibles entre ambos sistemas

En relación a la comunicación de computadores y redes de computadores, hay dos conceptos de suma importancia:

- Protocolos.
- Arquitectura de comunicaciones o arquitectura de protocolos.

Un **protocolo** se utiliza para comunicar entidades de diferentes sistemas. Lo que se comunica, cómo se comunica y cuándo se comunica, debe hacerse de acuerdo a unas convenciones entre las entidades involucradas. 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.

Arquitectura de protocolos:

Dada la complejidad de los sistemas, en la red hay que establecer una arquitectura de protocolos, que será la encargada de realizar las comunicaciones entre máquinas dividiendo las tareas en subtareas, en módulos, que contienen claves, mandatos, registros,... otros que ese encarguen de comprobar que las comunicaciones están bien establecidas,... En general, una estructura que sea capaz de responder de manera eficiente ante las diferencias de los sistemas que se están comunicando.

Arquitectura Multiprocesador: Sistemas Operativos Paralelos

Históricamente el computador se ha considerado como una unidad de procesamiento secuencial, donde toda instrucción se sucedía una tras otra, pero este concepto ha ido cambiando, pues hoy en día, por motivos de optimización, se utilizan nuevas técnicas de paralelismo (procesamiento de funciones en varios dispositivos a la vez). En particular el *Multiprocesamiento Simétrico (SMP) y los clusters.*

ARQUITECTURA SMP

Es útil ver donde encaja la arquitectura SMP dentro de las categorías de procesamiento paralelo. La forma más común de categorizar estos sistemas es la clasificación de sistemas de procesamiento paralelo introducida por Flynn. El cual propone las siguientes categorías de sistemas de computadores:

- **Única instrucción, único flujo de datos**: *(Single instruction single data (SISD) stream).* Un solo procesador ejecuta una única instrucción que opera sobre datos almacenados en una sola memoria.
- **Única instrucción, múltiples flujos de datos** : *(Single instruction multiple data (SIMD) stream).* Una única instrucción de máquina controla la ejecución simultánea de un número de elementos de proceso. Cada elemento de proceso tiene una memoria de datos asociada, de forma que cada instrucción se ejecuta en un conjunto de datos diferente a través de los diferentes procesadores. Los procesadores vectoriales y matriciales entran dentro de esta categoría.
- **Múltiples instrucciones, único flujo de datos** : *(Multiple instruction single data (MISD) stream).* Se transmite una secuencia de datos a un conjunto de procesadores, cada uno de los cuales ejecuta una secuencia de instrucciones diferente. Esta estructura nunca se ha implementado.
- **Múltiples instrucciones, múltiples flujos de datos**: *(Multiple instruction multiple data (MIMD) stream).* Un conjunto de procesadores ejecuta simultáneamente diferentes secuencias de instrucciones en diferentes conjuntos de datos.

Un sistema con esta última organización MIMD recibe el nombre de ***cluster***, donde si cada procesador accede a una misma memoria se le conoce como **multiprocesador de memoria compartida**. Y en estos se usa una estructura de maestro/esclavo para controlar y organizar, pero esto tiene algunas desventajas, por ejemplo un fallo en el maestro puede derrumbar todo el sistema, o las limitaciones en este pueden afectar a los que se encuentran subyugados.

ORGANIZACIÓN SMP

La organización de procesamiento multisimétrico permite que se planifiquen todas las tareas y se dividan en hilos previos a su procesamiento, cada proce-

sador contiene sus propias unidades, pero se encuentran unidas a una memoria compartida

DISEÑOS DE SISTEMAS OPERATIVOS MULTIPROCESADOR

El diseño de un SO permite encargarse de todas las peticiones sin que el usuario tenga que responder a estas, por lo tanto este diseño debe de tener unas características especiales como:

- **Procesos o hilos simultáneos concurrentes**. Las rutinas del núcleo necesitan ser reentrantes para permitir que varios procesadores ejecuten el mismo código del núcleo simultáneamente. Debido a que múltiples procesadores pueden ejecutar la misma o diferentes partes del código del núcleo, las tablas y la gestión de las estructuras del núcleo deben ser gestionas apropiadamente para impedir interbloqueos u operaciones inválidas.
- **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.
- **Sincronización**. Con múltiples procesos activos, que pueden acceder a espacios de direcciones compartidas o recursos compartidos de E/S, se debe tener cuidado en proporcionar una sincronización eficaz. La sincronización es un servicio que fuerza la exclusión mutua y el orden de los eventos.
- **Gestión de memoria**. La gestión de memoria en un multiprocesador debe tratar con todos los aspectos encontrados en las máquinas uniprocesador. Además, 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.