Lecturas 3 de Octubre

- **Estructura o Arquitectura del sistema y ejemplos de distintos SO. Micronúcleos (o microkernels).**

- **Estructura de los SO.**

Estructura del Sistema: Stallings 76-101

A medida que se han añadido más características a los SO y el hardware se ha vuelto más potente y versátil, han crecido el tamaño y la complejidad de los sistemas operativos.

El tamaño del SO, tiene cuatro grandes problemas, se entregan tarde de forma crónica, tienen fallos latentes que deben ser resueltos, el rendimiento no suele ser el esperado y es casi imposible construir un SO complejo invulnerable.

Para SO complejos, no es suficiente con la programación modular. La estructura jerárquica de un SO moderno separa sus funciones de acuerdo a las características.

Se presenta un modelo de SO jerárquico:

- Nivel 1: circuitos electrónicos.
- Nivel 2: conjunto de instrucciones del procesador.
- Nivel 3: concepto de procedimiento o subrutina.
- Nivel 4: introduce las interrupciones.

Estos cuatro niveles pertenecen al hardware del procesador.

- Nivel 5: proceso como programa.
- Nivel 6: trata los dispositivos de almacenamiento secundario del computador.
- Nivel 7: crea espacio de direcciones lógicas para los procesos.

El sistema operativo con los recursos de un único procesador.

- Nivel 8: comunicación de información y mensajes entre los procesos.
- Nivel 9: soporte de almacenamiento a largo plazo en ficheros con nombre.
- Nivel 10: proporciona acceso a los dispositivos externos utilizando interfaces estándar.
- Nivel 11: asociación entre los identificadores externos e internos de los recursos y objetos del sistema.
- Nivel 12: una utilidad completa para dar soporte a los procesos.
- Nivel 13: proporciona una interfaz del SO al usuario. Shell.

Desarrollos que han llevado a los SO modernos

Estos sistemas operativos modernos responden a nuevos desarrollos en hardware, nuevas aplicaciones y nuevas amenazas de seguridad.

Entre las causas de hardware principales se encuentran las máquinas multiprocesador, que han logrado incrementar la velocidad de la máquina.

Respecto a la seguridad, el acceso a Internet de los mismos es una de las principales amenazas.

- Arquitectura microkernel.
- Multihilo.
- Multiprocesamiento simétrico.
- SO distribuidos.
- Diseño Orientado a Objetos.

Por otro lado, una **arquitectura micronúcleo** asigna sólo unas pocas funciones esenciales al núcleo. Ciertos procesos proporcionan otros servicios del sistema operativo, algunas veces denominados servidores. Esta técnica desacopla el núcleo y el desarrollo del servidor.

Multithreading (multihilo) 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 y su propia área de datos para una pila.

Proceso: Es una colección de uno o más hilos y sus recursos de sistema asociados.

La técnica multithreading es útil para las aplicaciones que llevan a cabo un número de tareas independientes.

Para lograr mayor eficiencia y fiabilidad, una técnica consiste en emplear **multiprocesamiento simétrico**(SMP).

Características:

1. Tiene múltiples procesadores.
2. Estos procesadores comparten las mismas utilidades de memoria principal y de E/S.
3. Todos los procesadores pueden realizar las mismas funciones.

Ventajas:

- Rendimiento.
- Disponibilidad.
- Crecimiento incremental.
- Escalado.

La técnica multithreading y SMP son frecuentemente analizados juntos. Una característica atractiva de un SMP es que la existencia de múltiples procesadores

es transparente al usuario.

Descripción Global de Microsoft Windows

Historia

La historia de Windows comienza con un sistema operativo muy diferente, desarrollado por Microsoft para el primer computador personal IBM y conocido como MS-DOS o PC-DOS. La versión inicial apareció en 1981. Estaba compuesto por 4000 líneas de código fuente en ensamblador y ejecutaba en 8 Kbytes de memoria, utilizando el microprocesador Intel 8086.

Multitarea Monousuario

Una de las características más significativas de estos nuevos sistemas operativos es que, aunque están todavía pensados para dar soporte a un único usuario interactivo, se trata de sistemas operativos multitarea. Una segunda motivación para la multitarea es el aumento de la computación cliente/servidor. Una aplicación puede implicar a uno o más computadores personales y uno o más dispositivos de servidor. Para proporcionar la respuesta requerida, el SO necesita dar soporte al hardware de comunicación en tiempo real.

Organización del SO

Algunos elementos que forman parte del SO y que organizan el mismo son:

- Sistema ejecutivo.
- Núcleo.
- Capa de abstracción de hardware.
- Controladores de dispositivo.
- Gestión de ventanas y sistemas gráficos.
- Gestor de E/S.
- Gestor de caché.
- Gestor de memoria virtual.
- Gestor de procesos e hilos.
- Unidad de llamada de procedimiento local.

Procesos en Modo Usuario

- Procesos de sistema especiales.
- Procesos de servicio.
- Subsistemas de entorno.
- Aplicaciones de usuario.

Modelo cliente/servidor

- Simplifica el sistema ejecutivo.
- Mejora la fiabilidad.
- Proporciona a aplicaciones manera de comunicarse con el sistema ejecutivo.
- Base adecuada para computación distribuida.

Hilos y SMP

Características importantes de Windows:

- Rutinas del sistema se pueden ejecutar en cualquier procesador disponible.
- Windows permite uso múltiple de hilos de ejecución.
- Windows proporciona mecanismos para compartir datos y recursos entre procesos.

Sistemas UNIX Tradicionales

Historia

UNIX se desarrolló inicialmente en los laboratorios Bell y se hizo operacional en un PDP-7 en 1970. Este proyecto se encargó primero del desarrollo de CTSS y después de Multics.

Descripción

UNIX viene equipado con un conjunto de servicios de usuario e interfaces que se consideran parte del sistema. Estos se pueden agrupar en el shell, otro software de interfaz, y los componentes del compilador C. La capa externa está formada por las aplicaciones de usuario y la interfaz de usuario al compilador C.

Ejemplos de Sistemas UNIX Modernos:

1. SYSTEM V RELEASE 4
2. SOLARIS 9
3. 4.4BSD

Linux

Historia

Linux comenzó como una variante UNIX para la arquitectura del PC IBM (Intel 80386). Linus Torvalds, un estudiante finlandés de informática, escribió la versión inicial. Torvalds distribuyó por Internet una primera versión de Linux en 1991.

Estructura Modular

La mayoría de los núcleos Linux son monolíticos. Aunque Linux no utiliza una técnica de micronúcleo, logra muchas de las ventajas potenciales de esta técnica por medio de su arquitectura modular particular. Linux está estructurado como una colección de módulos, algunos de los cuales pueden cargarse y descargarse automáticamente bajo demanda. Estos bloques relativamente independientes se denominan módulos cargables. Esencialmente, un módulo es un fichero cuyo código puede enlazarse y desenlazarse con el núcleo en tiempo real.

Características:

- Enlace dinámico.
- Módulos apilables.

Ventajas:

1. Código común para módulos similares.
2. Núcleo puede asegurar que los módulos necesarios están presentes.

Algunos componentes del núcleo son:

- Señales.
- Llamadas al sistema.
- Procesos y planificador.
- Memoria virtual.
- Sistemas de ficheros.
- Controladores de dispositivos.
- Traps y fallos.

Micronúcleos: Stalling 176-181

Un **micronúcleo** es la pequeña parte central de un SO que proporciona las bases para extensiones modulares. El enfoque de micronúcleo se popularizó por su uso en el sistema operativo Mach. Proporciona un alto grado de flexibilidad y modularidad.

Arquitectura Micronúcleo

A finales de los años 50, los sistemas operativos monolíticos gozaban de falta de estructura, la que hacía insostenible a medida que los SO crecían. Por ello, se desarrollaron los SO por capas en los cuales las funciones se organizaban jerárquicamente.

Aunque los problemas permanecen incluso en el enfoque por capas, por ello apareció la filosofía del micronúcleo, la cual comparte que solamente las funciones absolutamente esenciales del SO están en el núcleo.

La arquitectura del micronúcleo reemplaza a la tradicional estructura vertical y estratificada en capas por una horizontal.

Beneficios de una organización Micronúcleo

El micronúcleo posee una interfaz uniforme en las peticiones realizadas por un proceso.

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 pueden eliminar para realizar una implementación más pequeña y eficiente.

Todo o gran parte del código específico del procesador está en el microkernel.

Rendimiento del Micronúcleo

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

Diseño del Micronúcleo

Gestión de memoria a bajo nivel:

El microkernel tiene que controlar el concepto de hardware de espacio de direcciones para hacer posible la implementación de protección a nivel de proceso.

Para comenzar, el núcleo asigna toda la memoria física disponible como recursos a un proceso base del sistema.

Un puerto es una cola de mensajes destinada a un proceso particular (un proceso puede tener múltiples puertos).

El paso de mensajes entre dos procesos que no tengan solapado el espacio de direcciones implica realizar una copia de memoria a memoria.

Gestión de E/S de interrupciones:

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

La transformación de las interrupciones en mensajes las debe realizar el microkernel, pero el microkernel no está relacionado con el manejo de interrupciones específico de los dispositivos.

Estructura de los SO y ejemplos: Silberchatz 52-62

Implementación del SO

Una vez diseñado el SO, debe implementarse, lo cual actualmente se hace en lenguajes de alto nivel como C o C++, lo que hace que el código pueda escribirse más rápido y es más fácil de entender y depurar, esto también mejora la portabilidad.

Linux está escrito principalmente en C y está disponible para una serie de CPU diferentes. Las desventajas residen en temas de velocidad y de espacio de almacenamiento.

Las mejoras de los rendimientos de los SO, son resultado de utilizar mejores estructuras de datos y mejores algoritmos.

Estructura del SO

La ingeniería de un sistema de un SO debe hacerse cuidadosamente. El método habitual es la división en componentes más pequeños, en lugar del sistema monolítico.

Estructura simple

Sistemas pequeños, simples y limitados, por ejemplos MS-DOS. Fue escrito para tener la mayor funcionalidad en el menor espacio posible. La división por capas no es cuidadosa y las aplicaciones pueden acceder directamente a las rutinas de E/S, haciéndolo muy vulnerable.

El kernel se divide en interfaces y controladores del dispositivo. El kernel está por debajo de las llamadas al sistema y por encima del hardware.

Estructura en niveles

Con el hardware apropiado, los SO pueden dividirse en más partes, teniendo más control sobre la computadora. Se propicia la ocultación de detalles.

El ejemplo más característico es la estructura en niveles. La ventaja de este método es la simplicidad y depuración. Los niveles superiores obtienen la información de los inferiores y se implementan a partir de ellos.

La principal dificultad es la forma de establecer esa jerarquía de capas o niveles.

Estas implementaciones tienden a ser menos eficientes, ya que cuando por ejemplo, se ejecuta una operación de E/S, cada nivel añade así una carga de trabajo al sistema que tarda más en ejecutarse.

Microkernels

A mediados de los 80, se desarrolló un SO, Mach, que implementaba el primer microkernel. Este método de estructura elimina todos los componentes no esenciales del kernel y los implementa como programas del sistema. Su función principal es proporcionar mecanismos de comunicación entre programa cliente y los distintos servicios.

Módulos

La mejor opción actualmente es el kernel modular, el kernel dispone de un conjunto de componentes fundamentales y enlaza dinámicamente los servicios adicionales.

El SO Solaris está organizado entorno a kernel modular que contiene.

1. Clases de planificación.
2. Sistemas de archivos.
3. Llamadas al sistema cargables.
4. Formatos ejecutables.
5. Módulos Streams.
6. Módulos misceláneos.
7. Controladores de bus y dispositivos.

También se puede encontrar una estructura híbrida, en la que se usa una técnica por niveles en la que uno de esos niveles es un microkernel.

Máquinas virtuales

La estructura por niveles tiene su conclusión lógica con el concepto de máquina virtual. La idea fundamental es la de abstraer el hardware de la computadora formando varios entornos de ejecución diferentes, creando así la ilusión de que cada entorno de ejecución está operando con su propia computadora privada.

Así el SO, puede crear la ilusión de que cada proceso tiene su propio procesador. Una de las principales dificultades del método son los sistemas de disco, no podemos crear siete unidades virtuales en tres discos, por ello surge el concepto de minidisco, obteniendo tantos minidiscos en el disco físico como sea necesario.

Implementación:

El concepto de máquina virtual resulta difícil de implementar.

La ventaja de ella, es el tiempo ya que la E/S virtual tarda mucho menos que en la real.

Beneficios:

Cada máquina virtual presenta protección completa de los diversos recursos del sistema. Pero no es posible la compartición directa de recursos. Para ello se han implementado dos métodos:

1. Compartir el minidisco.
2. Definir una red de máquinas virtuales, pudiendo cada una enviar información a través de la red de comunicaciones virtual.

Ejemplos de máquinas virtuales:

VMware:

Permite al sistema host ejecutar de forma concurrente varios sistemas operativos huésped diferentes como máquinas virtuales independientes.

Máquina virtual Java:

El JVM consta para ello de un cargador de clases y de un interprete de Java que ejecuta el código intermedio neutro. El JVM puede implementarse encima de un SO o bien como parte de un explorador web.