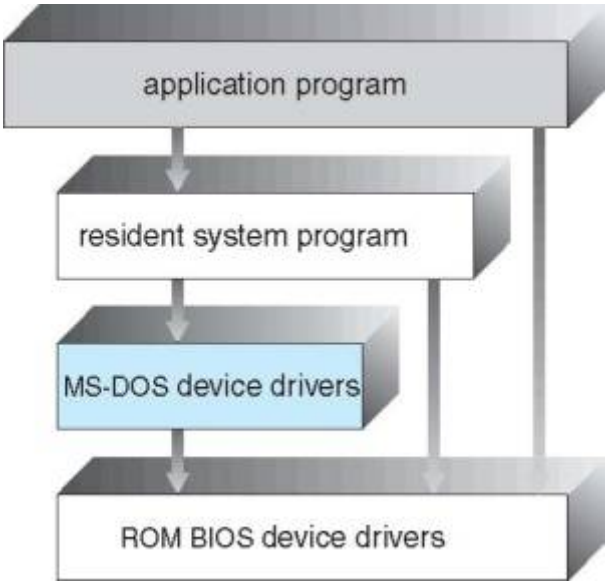
Estructura del sistema operativo

La ingeniería de un sistema tan grande y complejo como un sistema operativo moderno debe hacerse cuidadosamente para que el sistema funcione apropiadamente y pueda modificarse con facilidad. Un método habitual consiste en dividir la tarea en componentes más pequeños, en lugar de tener un sistema monolítico. Cada uno de estos módulos debe ser una parte bien definida del sistema, con entradas, salidas y funciones cuidadosamente especificadas.

Estructura simple

Muchos sistemas comerciales no tienen una estructura bien definida. Frecuentemente, tales sistemas operativos comienzan siendo sistemas pequeños, simples y limitados, y luego crecen más allá de su ámbito original; MS-DOS es un ejemplo de un sistema así. Originalmente, fue diseñado e implementado por unas pocas personas que no tenían idea de que iba a terminar siendo tan popular. Fue escrito para proporcionar la máxima funcionalidad en el menor espacio posible., por lo que no fue dividido en módulos de forma cuidadosa. La siguiente figura muestra su estructura.

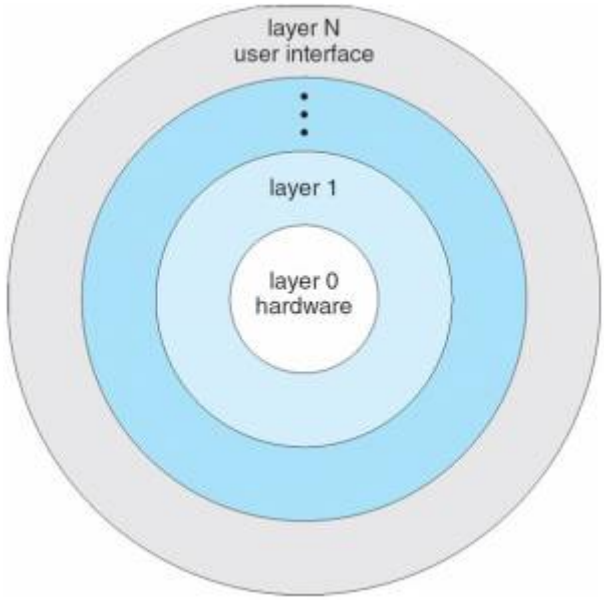


En MS-DOS, las interfaces y niveles de funcionalidad no están separados. Por ejemplo, los programas de aplicación pueden acceder a las rutinas básicas de E/S para escribir directamente en la pantalla y las unidades de disco. Tal libertad hace que MS-DOS sea vulnerable a programas erróneos (o maliciosos), lo que hace que el sistema completo falle cuando los programas de usuario fallan. Como en 8088 de Intel para el que fue escrito no proporciona un modo dual ni protección hardware, los diseñadores de MS-DOS no tuvieron más opción que dejar accesible el hardware base.

Estructura en niveles

Con el soporte de hardware apropiado, los sistemas operativos pueden dividirse en partes más pequeñas y más adecuadas que lo que permitían los sistemas originales MS-DOS o UNIX. El sistema operativo puede entonces mantener un control mucho mayor sobre la computadora y sobre las aplicaciones que hacen uso de dicha computadora. Los implementadores tienen más libertad para

cambiar el funcionamiento interno del sistema y crear sistemas operativos modulares. Con el método de diseño arriba-abajo, se determinan las características y la funcionalidad globales y se separan en componentes. La ocultación de los detalles a ojos de los niveles superiores también es importante, dado que deja libres a los programadores para implementar las rutinas de bajo nivel como prefieran, siempre que la interfaz externa de la rutina permanezca invariable y la propia rutina realice la tarea anunciada. Un sistema puede hacerse modular de muchas formas. Un posible método es mediante una estructura de niveles, en que el sistema operativo se divide en una serie de capas (niveles). El nivel inferior (nivel 0) es el hardware; el nivel superior (nivel N) es la interfaz de usuario. Esta estructura de niveles se ve en la siguiente figura.



Un nivel de un sistema operativo es una implementación de un objeto abstracto formado por una serie de datos y por las operaciones que permiten manipular dichos datos. Un nivel típico (por ejemplo, el nivel M) consta de estructuras de datos y de un conjunto de rutinas que los niveles superiores pueden invocar. A su vez el nivel M puede invocar operaciones sobre los niveles inferiores.

La principal ventaja del método de niveles es la simplicidad de construcción y depuración. Los niveles se seleccionan de modo que cada uno usa funciones (operaciones) y servicios de los niveles inferiores. Este método simplifica la depuración y la verificación del sistema. El primer nivel puede depurarse sin afectar al resto del sistema, dado que, por definición, sólo usa el hardware básico (que se supone correcto) para implementar sus funciones. Una vez que el primer nivel se ha depurado, puede suponerse correcto su funcionamiento mientras se depura el segundo nivel, etc. Si se encuentra un error durante la depuración de un determinado nivel, el error tendrá que estar localizado en dicho nivel, dado que los niveles inferiores a él ya se han depurado. Por tanto, el diseño e implementación del sistema se simplifican.

Cada nivel se implementa utilizando sólo las operaciones proporcionadas por los niveles inferiores. Un nivel no necesita saber cómo se implementan dichas operaciones; sólo necesita saber qué hacen esas operaciones. Por lo tanto cada nivel oculta a los niveles superiores la existencia de determinadas estructuras de datos, operaciones y hardware.

La principal dificultad con el método de niveles es la de definir apropiadamente los diferentes niveles. Dado que un nivel sólo puede usar los servicios de los niveles inferiores, es necesario realizar una planificación cuidadosa. Por ejemplo, el controlador de dispositivo para almacenamiento de reserva (espacio en disco usado por los algoritmos de memoria virtual) debe estar en un nivel inferior que las rutinas de gestión de memoria, dado que la gestión de memoria requiere la capacidad de usar

almacenamiento de reserva.

Microkernels

A medida que UNX se expandía, el kernel se hizo grande y difícil de gestionar. A mediados de los 80, los investigadores de la Universidad de Carnegie Mellon desarrollaron un sistema operativo denominado Mach que modularizaba el kernel usando lo que se denomina microkernel. Este método estructura el sistema operativo eliminando todos los componentes no esenciales del kernel e implementándolos como programas del sistema y de nivel de usuario; el resultado es un kernel más pequeño. No hay consenso en lo que se refiere a qué servicios deberían permanecer en el kernel y cuáles deberían implementarse en el espacio de usuario. Sin embargo, normalmente los microkernels proporcionan una gestión de la memoria y de los procesos mínima, además de un mecanismo de comunicaciones. La función principal del microkernel es proporcionar un mecanismo de comunicaciones entre el programa cliente y los distintos servicios que se ejecutan también en el espacio de usuario. La comunicación se proporciona mediante paso de mensajes. El programa cliente y el servicio nunca interactúan directamente, sino que se comunican de forma indirecta intercambiando mensajes con el microkernel. Otra ventaja es la facilidad para ampliar el sistema operativo. Todos los servicios nuevos se añaden al espacio de usuario y, en consecuencia, no requieren que se modifique el kernel.

From:
http://wiki.educabit.ar/ - **Wiki Sistemas**

Permanent link:
http://wiki.educabit.ar/doku.php?id=struc_so

Last update: **2025/09/11 22:48**