

Sistemas Operativos

Fundamentos de Sistemas Operativos

Índice

Pág.

1.1. Definición de sistema operativo	3
1.2. Objetivos de un sistema operativo	4
1.3. Evolución de los sistemas operativos	6
1.3.1. Procesamiento en serie	7
1.3.2. Sistemas por lotes	7
1.3.3. Multiprogramación	8
1.3.4. Sistema Operativo de tiempo compartido	8
1.3.5. Sistemas Paralelos	9
1.3.6. Sistemas Distribuidos	10
1.4. Sistemas Operativos Libres y propietarios	11
1.4.1. Freeware	12
1.4.2. Shareware	13
1.4.3. Software de Dominio Público	14
1.4.4. Software con Copyleft	15
Recursos complementarios	16
Bibliografía	16

Cuando una computadora nueva sale de la línea de montaje de fábrica, no puede hacer nada. El hardware necesita software para que funcione. ¿Estamos hablando de software de aplicaciones como procesamiento de texto o software de hoja de cálculo? Parcialmente. Pero un paquete de software de aplicaciones no se comunica directamente con el hardware.

Entonces, ¿qué sucede cuando se ejecuta un programa?. Bueno, un programa que se ejecuta hace una cosa muy sencilla: ejecuta instrucciones. Muchos millones (y hoy en día, incluso miles de millones) de veces por segundo, el procesador obtiene una instrucción de la memoria, la descodifica (es decir, averigua de qué instrucción se trata) y la ejecuta (es decir, hace lo que se supone que debe hacer, como sumar dos números, acceder a la memoria, comprobar una condición, saltar a una función, etc.). Una vez que ha terminado con esta instrucción, el procesador pasa a la siguiente, y así sucesivamente, hasta que el programa se completa. Así, acabamos de describir los fundamentos del modelo de computación de Von Neumann.

Hay una sección de software, de hecho, que es responsable de facilitar la ejecución de programas (incluso permitiendo aparentemente ejecutar muchos al mismo tiempo), permitiendo que los programas compartan memoria, permitiendo que los programas interactúen con los dispositivos, y otras cosas divertidas como esa. Ese conjunto de software se llama sistema operativo (SO), ya que se encarga de que el sistema funcione correctamente y de forma eficiente y fácil de usar.

1.1. Definición de sistema operativo

Un programa que actúa como intermediario entre el usuario de un ordenador y el hardware del mismo. El propósito de un sistema operativo es proporcionar un entorno en el que un usuario pueda ejecutar programas.

Todo esto ha marcado unos nuevos modos de producción y consumos culturales, donde la digitalización no sólo crea formatos transportables, de fácil almacenamiento, de calidad superior, sino que además este código permite.

Como se muestra en la Figura 1, entre el software de aplicaciones y el hardware hay una interfaz de software: un sistema operativo. Un sistema operativo es un conjunto de programas que se encuentra entre el software de aplicaciones y el hardware de la computadora. Conceptualmente, el software del sistema operativo es un intermediario entre el hardware y el software de aplicaciones.



Figura 1. Ubicación del Sistema Operativo

Por cierto, el término software del sistema a veces se usa indistintamente con el sistema operativo, pero software del sistema significa todos los programas relacionados con la coordinación de las operaciones de la computadora. El software del sistema incluye el sistema operativo, pero también incluye el software BIOS, controladores y programas de servicio.

1.2. Objetivos de un sistema operativo

Los sistemas operativos modernos generalmente tienen los siguientes tres objetivos principales. Los sistemas operativos generalmente logran estos

objetivos ejecutando procesos con privilegios bajos y proporcionando llamadas de servicio que invocan el kernel del sistema operativo en estado de privilegios altos.

Ocultar detalles de hardware mediante la creación de abstracción: Una abstracción es un software que oculta detalles de nivel inferior y proporciona un conjunto de funciones de nivel superior. Un sistema operativo transforma el mundo físico de dispositivos, instrucciones, memoria y tiempo en un mundo virtual que es el resultado de abstracciones creadas por el sistema operativo. Hay varias razones para la abstracción.

Primero , el código necesario para controlar los dispositivos periféricos no está estandarizado. Los sistemas operativos proporcionan subrutinas llamadas controladores de dispositivos que realizan operaciones en nombre de programas, por ejemplo, operaciones de entrada / salida.

Segundo, el sistema operativo introduce nuevas funciones a medida que abstrae el hardware. Por ejemplo, el sistema operativo introduce la abstracción de archivos para que los programas no tengan que trabajar con discos.

En tercer lugar , el sistema operativo transforma el hardware de la computadora en varias computadoras virtuales, cada una de las cuales pertenece a un programa diferente. Cada programa que se está ejecutando se denomina proceso. Cada proceso ve el hardware a través de la lente de la abstracción.

Cuarto, el sistema operativo puede reforzar la seguridad mediante la abstracción.

Para asignar recursos a procesos (administrar recursos): Un sistema operativo controla cómo los procesos (los agentes activos) pueden acceder a los recursos (entidades pasivas).

Proporcionar una interfaz de usuario agradable y eficaz: El usuario interactúa con los sistemas operativos a través de la interfaz de usuario y, por lo general, está interesado en la "apariencia" del sistema operativo. Los componentes más importantes de la interfaz de usuario son el intérprete de comandos, el sistema de archivos, la ayuda en línea y la integración de aplicaciones. La tendencia reciente ha sido hacia interfaces gráficas de usuario cada vez más integradas que abarcan las actividades de múltiples procesos en redes de computadoras.

Se pueden ver los sistemas operativos desde dos puntos de vista: administrador de recursos y máquinas extendidas .

1. Los sistemas operativos administran las diferentes partes del sistema de manera eficiente.

2. Los sistemas operativos proporcionan una máquina virtual a los usuarios que es más conveniente de usar.

1.3. Evolución de los sistemas operativos

Generalmente, los recursos necesarios son almacenamiento de archivos, CPU, memoria, dispositivos de entrada y salida, etc. El sistema operativo actúa como controlador de todos los recursos mencionados anteriormente y les asigna programas específicos ejecutados para realizar la tarea. Por lo tanto, el sistema operativo es un administrador de recursos que maneja el recurso como una vista de usuario y una vista del sistema. La evolución del sistema operativo está marcada desde la programación de tarjetas perforadas hasta máquinas de entrenamiento para hablar e interpretar cualquier idioma.

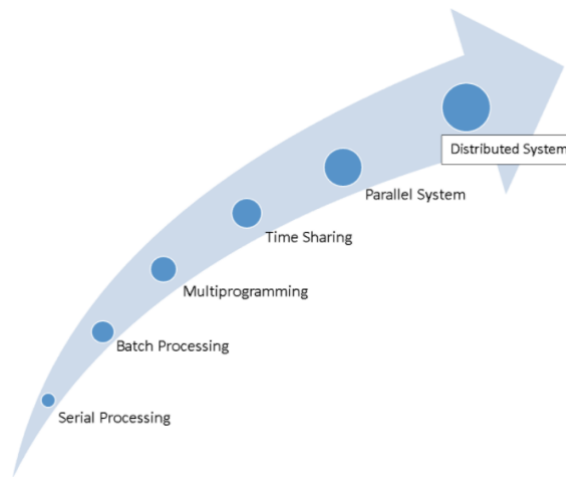


Figura 2. Evolución de los Sistemas Operativos

1.3.1. Procesamiento en serie

Se desarrolla entre 1940 y 1950, consta de programas incorporados por los componentes de hardware sin la implementación del sistema operativo. Los problemas aquí son la programación y el tiempo de configuración. El inicio de sesión del usuario para tiempo de máquina al perder el tiempo calculado. El tiempo de configuración está involucrado al cargar el compilador, guardar el programa compilado, el programa fuente, vincular y almacenar en búfer. Si ocurre algún error intermedio, el proceso comienza de nuevo.

1.3.2. El sistema por lotes

Se utiliza para mejorar la utilización y aplicación de las computadoras. Los trabajos se programaron y enviaron en tarjetas y cintas. Luego se ejecuta secuencialmente en los monitores utilizando Job Control Language. Las primeras computadoras se utilizan en el proceso de operación por lotes. El proceso de operación por lotes hizo que la computadora realizara lotes de trabajos sin ninguna pausa o parada. El programa se escribe en las tarjetas perforadas y luego se copia en la unidad de procesamiento de la cinta. Cuando la computadora completa un solo trabajo, inmediatamente comienza la siguiente tarea en la cinta. Los operadores profesionales están capacitados para comunicarse con la máquina donde los usuarios dejaron los trabajos y los recuperaron para recoger los resultados después de que se ejecuta el trabajo.

Aunque es incómodo para los usuarios, está hecho para mantener a una computadora tan ocupada en la medida de lo posible mediante la ejecución de un flujo de trabajos apalancado. La protección de la memoria no permite que el área de memoria que comprende el monitor se altere y el temporizador protege el trabajo de monopolizar el sistema. El procesador se mantiene inactivo cuando los dispositivos de entrada y salida están en uso debido a la mala utilización del tiempo de la CPU.

1.3.3. Multiprogramación

Este tipo de SO se utiliza para ejecutar más de un trabajo simultáneamente por un solo procesador de la CPU organizando los trabajos de forma que la CPU siempre tenga un trabajo que ejecutar.

El concepto de multiprogramación se describe como sigue: Todos los trabajos que entran en el sistema se almacenan en el pool de trabajos(en disco). El sistema operativo carga un conjunto de trabajos de la reserva de trabajos en la memoria principal y comienza a ejecutarlos. Durante la ejecución, el trabajo puede tener que esperar a que se complete alguna tarea, como una operación de E/S. En un sistema de multiprogramación, el sistema operativo simplemente cambia a otro trabajo y lo ejecuta. Cuando ese trabajo necesita esperar, la CPU cambia a otro trabajo, y así sucesivamente. Cuando el primer trabajo termina de esperar, recupera la CPU. Mientras haya al menos un trabajo que necesite ejecutarse, la CPU nunca está inactiva.

Los sistemas operativos de multiprogramación utilizan el mecanismo de programación de trabajos y programación de la CPU

1.3.4. Sistema operativo de tiempo compartido

Se utiliza para desarrollar los sistemas de lotes sustitutos. El usuario se comunica directamente con la computadora imprimiendo puertos como un teletipo eléctrico. Un ejemplo es que pocos usuarios compartan la computadora

instantáneamente y pasan una fracción de segundo en cada trabajo antes de comenzar con el siguiente. Un servidor puede actuar instantáneamente en los procesos de muchos usuarios creando la iteración cuando estaban recibiendo toda su atención. Los sistemas de tiempo compartido son utilizados por múltiples programas para aplicarlos al sistema informático al compartir el sistema de forma interactiva.

La multiprogramación se utiliza para gestionar múltiples trabajos comunicativos. El tiempo del procesador se comparte entre varios usuarios y muchos usuarios pueden acceder simultáneamente al sistema a través de terminales. Ej. Los puertos de impresión necesitan programas con la interfaz de usuario de línea de comandos, donde el usuario ha escrito respuestas a comandos o comandos escritos.

1.3.5. Sistemas paralelos

Los sistemas operativos multiprocesador también se conocen como sistemas operativos paralelos o sistemas operativos estrechamente acoplados. Estos sistemas operativos tienen más de un procesador en estrecha comunicación que comparte el bus del ordenador, el reloj y a veces la memoria y los dispositivos periféricos. Ejecuta múltiples trabajos al mismo tiempo y hace que el procesamiento sea más rápido. más rápido.

Los sistemas multiprocesador tienen tres ventajas principales:

1. **Aumento del rendimiento:** Al aumentar el número de procesadores, el sistema realiza más trabajo en menos tiempo.

El ratio de velocidad con N procesadores es menor que N.

2. **Economía de escala:** Los sistemas multiprocesadores pueden ahorrar más dinero que los sistemas de un solo procesador, porque pueden compartir periféricos, almacenamiento masivo y fuentes de alimentación.

3. **Mayor fiabilidad:** Si un procesador falla en su tarea, cada uno de los procesadores restantes debe una parte del trabajo del procesador que ha fallado. El fallo de un procesador no detendrá el sistema, sólo lo ralentizará. ralentizarlo.

La capacidad de seguir prestando un servicio proporcional al nivel del hardware superviviente se denomina graceful degradación. Los sistemas diseñados para una degradación gradual se denominan tolerantes a fallos.

1.3.6. Sistemas Distribuidos

En un sistema distribuido, las diferentes máquinas están conectadas en red y cada una de ellas tiene su propio procesador y su propia memoria local. En este sistema, los sistemas operativos de todas las máquinas trabajan juntos para gestionar los recursos colectivos de la red. de la red.

Se puede clasificar en dos categorías:

1. Sistemas cliente-servidor
2. Sistemas Peer-to-Peer

Algunas de las ventajas de los sistemas distribuidos son: Compartición de recursos, Aumento de la velocidad de cálculo - reparto de la carga, Fiabilidad, Comunicaciones, Requiere una infraestructura de red (Redes de área local (LAN) o redes de área amplia (WAN)).

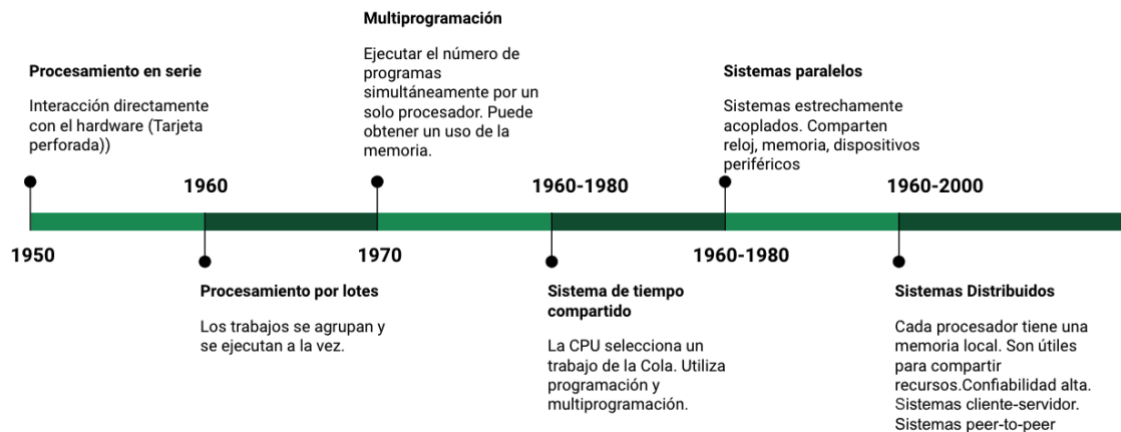


Figura 3. Evolución de los Sistemas Operativos con respecto a fechas

1.4. Sistemas Operativos Libres y propietarios

El término fuente abierta se refiere al software cuyo código fuente, el medio en el que los programadores crean y modifican el software, está disponible gratuitamente en Internet; por el contrario, el código fuente del software comercial propietario suele ser un secreto muy bien guardado.

El ejemplo más conocido de software de código abierto es el sistema operativo Linux, pero existen productos de software de código abierto disponibles para todos los propósitos imaginables.

El software de código abierto se distribuye bajo una variedad de términos de licencia, pero casi todos tienen dos cosas en común: el software se puede usar sin pagar una tarifa de licencia y cualquiera puede modificar el software para agregar capacidades no previstas por sus creadores.

Un estándar es una especificación tecnológica cuyos detalles están ampliamente disponibles, lo que permite a muchas empresas crear productos que funcionarán indistintamente y serán compatibles entre sí. Cualquier producto de tecnología moderna se basa en miles de estándares en su diseño, incluso la gasolina que pone en su automóvil se mezcla para cumplir con varias

especificaciones muy detalladas en las que confían los diseñadores del automóvil.

Para que un estándar se considere un estándar abierto , la especificación y los derechos para implementarlo deben estar disponibles gratuitamente para cualquier persona sin firmar acuerdos de confidencialidad o pagar regalías. El mejor ejemplo de estándares abiertos en funcionamiento es Internet: prácticamente todas las especificaciones tecnológicas de las que depende son abiertas, al igual que el proceso para definir nuevas.

Una interfaz de programación de aplicaciones (API) es una característica de una aplicación de software que permite que otro software interactúe con ella, invocando automáticamente su funcionalidad e intercambiando datos con ella. La definición de API es una forma de estándar tecnológico. El término API abierta aún no tiene una definición universalmente aceptada, pero generalmente se espera que sea "abierta" de la misma manera que un estándar abierto.

El tema común de "apertura" en las definiciones anteriores es la capacidad de diversas partes para crear tecnología que interopere. Al evaluar las necesidades de software actuales y previstas de su organización, considere la capacidad de una solución para interoperar como un criterio importante. Para ampliar el valor de su inversión en tecnología, seleccione una solución de software que se base en estándares abiertos y API que faciliten la interoperabilidad y tenga la capacidad de integración directa entre los productos de varios proveedores.

Existen otros conceptos que son necesarios estudiarlos:

1.4.1. Freeware

El software gratuito es un software que está *disponible de forma gratuita* . Un usuario puede descargar software gratuito de internet y

usarlo. Estos softwares *no brindan libertad para modificar, compartir y estudiar el programa* como en el software de código abierto. El software gratuito es de *código cerrado*.

Los usuarios prefieren el software gratuito por las siguientes razones:

- Disponible sin costo
- Se puede distribuir sin costo

Algunos ejemplos de software gratuito son:

- Yahoo Messenger
- Google Talk
- MSN Messenger

1.4.2. Shareware

El software Shareware es un software que se *distribuye gratuitamente a los usuarios a modo de prueba*. Hay un *límite de tiempo* incorporado en el software (por ejemplo, gratis por 30 días o 2 meses). A medida que se supera el límite de tiempo, se desactivará. Para usarlo después del límite de tiempo, debe pagar el software.

Los usuarios prefieren shareware por las siguientes razones:

- Disponible sin costo
- ayuda a conocer el producto antes de comprarlo

Algunos ejemplos de software gratuito son:

- Adobe acrobat 8 profesional
- PHP Debugger 2.1.3.3
- Winzip

Shareware son de los siguientes tipos:

- **Adware** : contiene anuncios para generar ingresos para los desarrolladores
- **Donationware** : el pago es opcional
- **Nagware** : le recuerda al usuario que compre la licencia o el software
- **Freemium** : gratis para no premium pero de costo para características premium
- **Demoware** - versión de demostración. Se clasifica además como crippleware y trialware.
- **Crippleware** : algunas funciones están deshabilitadas dentro del límite de tiempo
- **Software de prueba** : todas las funciones están disponibles bajo límite de tiempo

1.4.3. Software de dominio público

El software de dominio público es aquel que no tiene derechos de autor. Si el código fuente es de dominio público, se trata de un caso especial de software libre sin copyleft, lo que significa que algunas copias o versiones modificadas pueden no ser libres en absoluto.

En algunos casos, un programa ejecutable puede ser de dominio público pero no disponer libremente del código fuente. En ese caso no es software libre, porque el software libre requiere accesibilidad al código fuente. Por otro lado, la mayoría del software libre no está en el dominio público sino bajo los derechos de autor, y los titulares de esos derechos han dado el permiso legal para que todos puedan utilizarlo en libertad, usando una licencia de software libre.

Algunas personas utilizan el término «dominio público» de manera imprecisa queriendo decir «libre» o «gratuito». Sin embargo, «dominio público» es un término jurídico cuyo significado preciso es «sin derechos de autor».

Para ser lo más claro posible, recomendamos el uso del término «dominio público» para expresar solamente este significado, y el uso de las otras expresiones para transmitir sus significados correspondientes.

En el marco del Convenio de Berna, que la mayoría de los países han firmado, todo lo que se escribe queda automáticamente bajo el dominio de los derechos de autor, inclusive los programas informáticos. Por lo tanto, si usted quiere que un programa que ha escrito esté disponible en el dominio público, debe tomar algunas medidas legales para renunciar a esos derechos; de lo contrario el programa quedará sujeto a los derechos de autor.

1.4.4. Software con copyleft

El software con copyleft es software libre cuyos términos de distribución garantizan que todas las copias de todas las versiones tengan aproximadamente los mismos términos de distribución. Esto significa, por ejemplo, que las licencias copyleft generalmente no permiten que terceros le agreguen requisitos adicionales al software (aunque puede estar permitido agregar un conjunto limitado de requisitos que se consideran seguros) y exigen que el código fuente esté disponible. Esto tutela el programa y sus versiones modificadas contra algunas de las formas más comunes de convertirlo en software privativo.

Copyleft es un concepto general, para poner un programa bajo copyleft, es necesario adoptar un conjunto específico de cláusulas para la distribución. Existen varias maneras de redactar las cláusulas de copyleft, por lo que en principio pueden existir muchas licencias libres con copyleft. Sin embargo, en la práctica, para casi todo el software con copyleft se usa la Licencia Pública General de GNU (*GNU General Public License*). Generalmente dos licencias diferentes con copyleft son «incompatibles», lo cual significa que es ilegal combinar el código que está bajo un tipo de licencia con el código que está bajo otro tipo de licencia; por eso es bueno para la comunidad usar una sola licencia con copyleft.

Recursos complementarios

- Video sobre los sistemas operativos modernos



<https://www.youtube.com/watch?v=Oa9aWdcCC4o>



<https://www.youtube.com/watch?v=FroR3fqxK34>

Bibliografía

A. S. Tanenbaum, Sistemas operativos modernos. Pearson Educación, 2003.