Sistema operativos

Saltar a: navegación, búsqueda Interacción entre el SO con el resto de las partes. Estimación del uso de sistemas operativos según una muestra de computadoras con acceso a Internet en noviembre de 2009 (fuente: W3counter).

Un sistema operativo (SO) es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, y se ejecuta en modo privilegiado respecto de los restantes.

Nótese que es un error común muy extendido denominar al conjunto completo de herramientas sistema operativo, es decir, la inclusión en el mismo término de programas como el explorador de ficheros, el navegador web y todo tipo de herramientas que permiten la interacción con el sistema operativo, también llamado núcleo o kernel. Esta identidad entre kernel y sistema operativo es solo cierta si el núcleo es monolítico. Otro ejemplo para comprender esta diferencia se encuentra en la plataforma Amiga, donde el entorno gráfico de usuario se distribuía por separado, de modo que, también podía reemplazarse por otro, como era el caso de directory Opus o incluso manejarlo arrancando con una línea de comandos y el sistema gráfico. De este modo, al arrancar un Amiga, comenzaba a funcionar con el propio sistema operativo que llevaba incluido en una ROM, por lo que era cuestión del usuario decidir si necesitaba un entorno gráfico para manejar el sistema operativo o simplemente otra aplicación. Uno de los más prominentes ejemplos de esta diferencia, es el núcleo Linux, usado en las llamadas distribuciones Linux, ya que al estar también basadas en Unix, proporcionan un sistema de funcionamiento similar. Este error de precisión, se debe a la modernización de la informática llevada a cabo a finales de los 80, cuando la filosofía de estructura básica de funcionamiento de los grandes computadores se rediseñó a fin de llevarla a los hogares y facilitar su uso, cambiando el concepto de computador multiusuario, (muchos usuarios al mismo tiempo) por un sistema monousuario (únicamente un usuario al mismo tiempo) más sencillo de gestionar. de dicha modernización, cuando los Amiga fueron bautizados con el sobrenombre de Video Toasters por su capacidad para la Edición de vídeo en entorno multitarea round robin, con gestión de miles de colores e interfaces intuitivos para diseño en 3D.

Uno de los propósitos del sistema operativo que gestiona el núcleo intermediario consiste en gestionar los recursos de localización y protección de acceso del hardware, hecho que alivia a los programadores de aplicaciones de tener que tratar con estos detalles. La mayoría de aparatos electrónicos que utilizan microprocesadores para funcionar, llevan incorporado un sistema operativo. (teléfonos móviles, reproductores de DVD, computadoras, radios, enrutadores, etc). En cuyo caso, son manejados mediante una Interfaz Gráfica de Usuario, un gestor de ventanas o un entorno de escritorio, si es un celular, mediante una consola o control remoto si es un DVD y, mediante una línea de comandos o navegador web si es un enrutador.

Perspectiva histórica

Los primeros sistemas (1945-1955) eran grandes máquinas operadas desde la consola maestra por los programadores. Durante la década siguiente (1955-1965) se llevaron a cabo avances en el hardware: lectoras de tarjetas, impresoras, cintas magnéticas, etc. Esto a su vez provocó un avance en el software: compiladores, ensambladores, cargadores, manejadores de dispositivos, etc.

A finales de los años 1980, una computadora Commodore Amiga equipada con una aceleradora Video

Toaster era capaz de producir efectos comparados a sistemas dedicados que costaban el triple. Un Video Toaster junto a Lightwave ayudó a producir muchos programas de televisión y películas, entre las que se incluyen Babylon 5, Seaquest DSV y Terminator II.

Problemas de explotación y soluciones iniciales

El problema principal de los primeros sistemas era la baja utilización de los mismos, la primera solución fue poner un operador profesional que lo manejase, con lo que se eliminaron las hojas de reserva, se ahorró tiempo y se aumentó la velocidad.

Para ello, los trabajos se agrupaban de forma manual en lotes mediante lo que se conoce como procesamiento por lotes (batch) sin automatizar.

Monitores residentes

Un monitor residente estaba compuesto por un cargador, un Intérprete de comandos y un Controlador (drivers) para el manejo de entrada/salida.

Según fue avanzando la complejidad de los programas, fue necesario implementar soluciones que automatizaran la organización de tareas sin necesidad de un operador. Debido a ello se crearon los monitores residentes: programas que residían en memoria y que gestionaban la ejecución de una cola de trabajos.

Sistemas con almacenamiento temporal de E/S

Los avances en el hardware crearon el soporte de interrupciones y posteriormente se llevó a cabo un intento de solución más avanzado: solapar la E/S de un trabajo con sus propios cálculos, por lo que se creó el sistema de búfers con el siguiente funcionamiento:

- Un programa escribe su salida en un área de memoria (búfer 1).
- El monitor residente inicia la salida desde el buffer y el programa de aplicación calcula depositando la salida en el buffer 2.
- La salida desde el buffer 1 termina y el nuevo cálculo también.
- Se inicia la salida desde el buffer 2 y otro nuevo cálculo dirige su salida al buffer 1.
- El proceso se puede repetir de nuevo.

Los problemas surgen si hay muchas más operaciones de cálculo que de E/S (limitado por la CPU) o si por el contrario hay muchas más operaciones de E/S que cálculo (limitado por la E/S).

Spoolers

Hace aparición el disco magnético con lo que surgen nuevas soluciones a los problemas de rendimiento. Se eliminan las cintas magnéticas para el volcado previo de los datos de dispositivos lentos y se sustituyen por discos (un disco puede simular varias cintas). Debido al solapamiento del cálculo de un trabajo con la E/S de otro trabajo se crean tablas en el disco para diferentes tareas, lo

que se conoce como Spool (Simultaneous Peripherial Operation On-Line).

Sistemas operativos multiprogramados

Surge un nuevo avance en el hardware: el hardware con protección de memoria. Lo que ofrece nuevas soluciones a los problemas de rendimiento:

- Se solapa el cálculo de unos trabajos con la entrada/salida de otros trabajos.
- Se pueden mantener en memoria varios programas.
- Se asigna el uso de la CPU a los diferentes programas en memoria.

Debido a los cambios anteriores, se producen cambios en el monitor residente, con lo que éste debe abordar nuevas tareas, naciendo lo que se denomina como Sistemas Operativos multiprogramados, los cuales cumplen con las siguientes funciones:

- Administrar la memoria.
- Gestionar el uso de la CPU (planificación).
- Administrar el uso de los dispositivos de E/S.

Cuando desempeña esas tareas, el monitor residente se transforma en un sistema operativo multiprogramado.

Llamadas al sistema operativo

Definición breve: llamadas que ejecutan los programas de aplicación para pedir algún servicio al SO.

Cada SO implementa un conjunto propio de llamadas al sistema. Ese conjunto de llamadas es la interfaz del SO frente a las aplicaciones. Constituyen el lenguaje que deben usar las aplicaciones para comunicarse con el SO. Por ello si cambiamos de SO, y abrimos un programa diseñado para trabajar sobre el anterior, en general el programa no funcionará, a no ser que el nuevo SO tenga la misma interfaz. Para ello:

- Las llamadas correspondientes deben tener el mismo formato.
- Cada llamada al nuevo SO tiene que dar los mismos resultados que la correspondiente del anterior.

Modos de ejecución en un CPU

Las aplicaciones no deben poder usar todas las instrucciones de la CPU. No obstante el Sistema Operativo, tiene que poder utilizar todo el juego de instrucciones del CPU. Por ello, una CPU debe tener (al menos) dos modos de operación diferentes:

- Modo usuario: el CPU podrá ejecutar sólo las instrucciones del juego restringido de las aplicaciones.
- Modo supervisor: la CPU debe poder ejecutar el juego completo de instrucciones.

Llamadas al sistema

Una aplicación, normalmente no sabe dónde está situada la rutina de servicio de la llamada. Por lo que si ésta se codifica como una llamada de función, cualquier cambio en el S.O. haría que hubiera que reconstruir la aplicación.

Pero lo más importante es que una llamada de función no cambia el modo de ejecución de la CPU. Con lo que hay que conseguir llamar a la rutina de servicio, sin tener que conocer su ubicación, y hacer que se fuerce un cambio de modo de operación de la CPU en la llamada (y la recuperación del modo anterior en el retorno).

Esto se hace utilizando instrucciones máquina diseñadas específicamente para este cometido, distintas de las que se usan para las llamadas de función. Bibliotecas de interfaz de llamadas al sistema

Las llamadas al sistema no siempre tienen una expresión sencilla en los lenguajes de alto nivel, por ello se crean las bibliotecas de interfaz, que son bibliotecas de funciones que pueden usarse para efectuar llamadas al sistema. Las hay para distintos lenguajes de programación.

La aplicación llama a una función de la biblioteca de interfaz (mediante una llamada normal) y esa función es la que realmente hace la llamada al sistema. Interrupciones y excepciones

El SO ocupa una posición intermedia entre los programas de aplicación y el hardware. No se limita a utilizar el hardware a petición de las aplicaciones ya que hay situaciones en las que es el hardware el que necesita que se ejecute código del SO. En tales situaciones el hardware debe poder llamar al sistema, pudiendo deberse estas llamadas a dos condiciones:

- Algún dispositivo de E/S necesita atención.
- Se ha producido una situación de error al intentar ejecutar una instrucción del programa (normalmente de la aplicación).

En ambos casos, la acción realizada no está ordenada por el programa de aplicación, es decir, no figura en el programa.

Según los dos casos anteriores tenemos las interrupciones y la excepciones:

- Interrupción: señal que envía un dispositivo de E/S a la CPU para indicar que la operación de la que se estaba ocupando, ya ha terminado.
- Excepción: una situación de error detectada por la CPU mientras ejecutaba una instrucción, que requiere tratamiento por parte del SO.

Tratamiento de las interrupciones

Una interrupción se trata en todo caso, después de terminar la ejecución de la instrucción en curso.

El tratamiento depende de cuál sea el dispositivo de E/S que ha causado la interrupción, ante la cual debe poder identificar el dispositivo que la ha causado.

La ventaja de este procedimiento es que no se tiene que perder tiempo ejecutando continuamente rutinas para consultar el estado del periférico. El inconveniente es que el dispositivo debe tener los circuitos electrónicos necesarios para acceder al sistema de interrupciones del computador.

Importancia de las interrupciones

El mecanismo de tratamiento de las interrupciones permite al SO utilizar la CPU en servicio de una aplicación, mientras otra permanece a la espera de que concluya una operación en un dispositivo de E/S.

El hardware se encarga de avisar al SO cuando el dispositivo de E/S ha terminado y el SO puede intervenir entonces, si es conveniente, para hacer que el programa que estaba esperando por el dispositivo, se continúe ejecutando.

En ciertos intervalos de tiempo puede convenir no aceptar señales de interrupción. Por ello las interrupciones pueden inhibirse por programa (aunque esto no deben poder hacerlo las mismas).

Un ejemplo de sincronismo por interrupción es el almacenamiento de caracteres introducidos mediante el teclado. Cuando se introduce un carácter, se codifica en el registro de datos del dispositivo y además se activa un bit del registro de estado quien crea una interrupción en el hardware. El procesador deja temporalmente la tarea que estaba completando y ejecuta la rutina de atención a la interrupción correspondiente. El teclado almacena el carácter en el vector de memoria intermedia (también llamado buffer) asociada al teclado y despierta el proceso que había en el estado de espera de la operación de entrada/salida. Excepciones

Cuando la CPU intenta ejecutar una instrucción incorrectamente construida, la unidad de control lanza una excepción para permitir al SO ejecutar el tratamiento adecuado. Al contrario que en una interrupción, la instrucción en curso es abortada. Las excepciones al igual que las interrupciones deben estar identificadas. Clases de excepciones

Las instrucciones de un programa pueden estar mal construidas por diversas razones:

- El código de operación puede ser incorrecto.
- Se intenta realizar alguna operación no definida, como dividir por cero.
- La instrucción puede no estar permitida en el modo de ejecución actual.
- La dirección de algún operando puede ser incorrecta o se intenta violar alguno de sus permisos de uso.

Importancia de las excepciones

El mecanismo de tratamiento de las excepciones es esencial para impedir, junto a los modos de ejecución de la CPU y los mecanismos de protección de la memoria, que las aplicaciones realicen operaciones que no les están permitidas. En cualquier caso, el tratamiento específico de una excepción lo realiza el SO.

Como en el caso de las interrupciones, el hardware se limita a dejar el control al SO, y éste es el que trata la situación como convenga.

Es bastante frecuente que el tratamiento de una excepción no retorne al programa que se estaba

ejecutando cuando se produjo la excepción, sino que el SO aborte la ejecución de ese programa. Este factor depende de la pericia del programador para controlar la excepción adecuadamente. Componentes de un sistema operativo Componentes del Sistema Operativo. Gestión de procesos

Un proceso es simplemente, un programa en ejecución que necesita recursos para realizar su tarea: tiempo de CPU, memoria, archivos y dispositivos de E/S. El SO es el responsable de:

- Crear y destruir los procesos.
- Parar y reanudar los procesos.
- Ofrecer mecanismos para que se comuniquen y sincronicen.

La gestión de procesos podría ser similar al trabajo de oficina. Se puede tener una lista de tareas a realizar y a estas fijarles prioridades alta, media, baja por ejemplo. Debemos comenzar haciendo las tareas de prioridad alta primero y cuando se terminen seguir con las de prioridad media y después las de baja. Una vez realizada la tarea se tacha. Esto puede traer un problema que las tareas de baja prioridad pueden que nunca lleguen a ejecutarse. y permanezcan en la lista para siempre. Para solucionar esto, se puede asignar alta prioridad a las tareas más antiguas. [editar] Gestión de la memoria principal

La Memoria es una gran tabla de palabras o bytes que se referencian cada una mediante una dirección única. Este almacén de datos de rápido accesos es compartido por la CPU y los dispositivos de E/S, es volátil y pierde su contenido en los fallos del sistema. El SO es el responsable de:

- Conocer qué partes de la memoria están siendo utilizadas y por quién.
- Decidir qué procesos se cargarán en memoria cuando haya espacio disponible.
- Asignar y reclamar espacio de memoria cuando sea necesario.

Gestión del almacenamiento secundario

Un sistema de almacenamiento secundario es necesario, ya que la memoria principal (almacenamiento primario) es volátil y además muy pequeña para almacenar todos los programas y datos. También es necesario mantener los datos que no convenga mantener en la memoria principal. El SO se encarga de:

- Planificar los discos.
- Gestionar el espacio libre.
- Asignar el almacenamiento.
- Verificar que los datos se guarden en orden

El sistema de entrada y salida

Consiste en un sistema de almacenamiento temporal (caché), una interfaz de manejadores de dispositivos y otra para dispositivos concretos. El sistema operativo debe gestionar el almacenamiento temporal de E/S y servir las interrupciones de los dispositivos de E/S.

Sistema de archivos Artículo principal: Sistema de archivos.

Los archivos son colecciones de información relacionada, definidas por sus creadores. Éstos almacenan programas (en código fuente y objeto) y datos tales como imágenes, textos, información de bases de datos, etc. El SO es responsable de:

- Construir y eliminar archivos y directorios.
- Ofrecer funciones para manipular archivos y directorios.
- Establecer la correspondencia entre archivos y unidades de almacenamiento.
- Realizar copias de seguridad de archivos.

Existen diferentes sistemas de archivos, es decir, existen diferentes formas de organizar la información que se almacena en las memorias (normalmente discos) de los ordenadores. Por ejemplo, existen los sistemas de archivos FAT, FAT32, EXT3, NTFS, XFS, etc.

Desde el punto de vista del usuario estas diferencias pueden parecer insignificantes a primera vista, sin embargo, existen diferencias muy importantes. Por ejemplo, los sistemas de ficheros FAT32 y NTFS, que se utilizan fundamentalmente en sistemas operativos de Microsoft, tienen una gran diferencia para un usuario que utilice una base de datos con bastante información ya que el tamaño máximo de un fichero con un sistema de archivos FAT32 está limitado a 4 gigabytes, sin embargo, en un sistema NTFS el tamaño es considerablemente mayor. Sistemas de protección

Mecanismo que controla el acceso de los programas o los usuarios a los recursos del sistema. El SO se encarga de:

- Distinguir entre uso autorizado y no autorizado.
- Especificar los controles de seguridad a realizar.
- Forzar el uso de estos mecanismos de protección.

Sistema de comunicaciones

Para mantener las comunicaciones con otros sistemas es necesario poder controlar el envío y recepción de información a través de las interfaces de red. También hay que crear y mantener puntos de comunicación que sirvan a las aplicaciones para enviar y recibir información, y crear y mantener conexiones virtuales entre aplicaciones que están ejecutándose localmente y otras que lo hacen remotamente. Programas de sistema

Son aplicaciones de utilidad que se suministran con el SO pero no forman parte de él. Ofrecen un entorno útil para el desarrollo y ejecución de programas, siendo algunas de las tareas que realizan:

- Manipulación y modificación de archivos.
- Información del estado del sistema.
- Soporte a lenguajes de programación.
- Comunicaciones.

Gestor de recursos

Como gestor de recursos, el sistema operativo administra:

- La unidad central de procesamiento (donde está alojado el microprocesador).
- Los dispositivos de entrada y salida.
- La memoria principal (o de acceso directo).
- Los discos (o memoria secundaria).
- Los procesos (o programas en ejecución).
- Y en general todos los recursos del sistema.

Clasificación

Administración de tareas

- Monotarea: Solamente permite ejecutar un proceso (aparte de los procesos del propio SO) en un momento dado. Una vez que empieza a ejecutar un proceso, continuará haciéndolo hasta su finalización y/o interrupción.
- Multitarea: Es capaz de ejecutar varios procesos al mismo tiempo. Este tipo de SO. normalmente asigna los recursos disponibles (CPU, memoria, periféricos) de forma alternada a los procesos que los solicitan, de manera que el usuario percibe que todos funcionan a la vez, de forma concurrente.

Administración de usuarios

- Monousuario: Si sólo permite ejecutar los programas de un usuario al mismo tiempo.
- Multiusuario: Si permite que varios usuarios ejecuten simultáneamente sus programas, accediendo a la vez a los recursos de la computadora. Normalmente estos sistemas operativos utilizan métodos de protección de datos, de manera que un programa no pueda usar o cambiar los datos de otro usuario.

Manejo de recursos

- Centralizado: Si permite usar los recursos de una sola computadora.
- Distribuido: Si permite utilizar los recursos (memoria, CPU, disco, periféricos...) de más de una computadora al mismo tiempo.

Ejemplos de sistemas operativos para PC

- Debian - GNU/Linux
- Ubuntu - GNU/Linux
- Mandriva - GNU/Linux
- Sabayon - GNU/Linux
- Fedora - GNU/Linux
- Mac OS Lion - Unix
- Solaris - Unix
- Google Chrome OS - GNU/Linux

- Windows 10

Ejemplos de sistemas operativos para dispositivos móviles

- Symbian
- Android
- iOS
- Windows Phone
- BlackBerry OS
- WebOS
- Bada

Referencias

- 1. ↑ Cfr., entre otros: Tanenbaum, A. (1992) Modern Operating Systems, Englewood Cliffs: Prentice-Hall; Haldar, S. y Aravind, A. A. (2010) Operating Systems, Pearson Education India, pp. 12 y ss.; Turner, R. W. (1986) Operating Systems: design and imnplementation, MacMillan.
- 2. ↑ Cisco Systems (1992). cisco.com (ed.): «Fundamentos de UNIX» (certificación). Consultado el 10 de febrero de 2010.
- 3. ↑ «Washington Post - Debut de Windows 95» (en inglés).
- 4. ↑ toastytech.com (ed.): «BeOS 5.0 Personal Edition» (en inglés) (2010). Consultado el 10 de febrero de 2010.
- 5. ↑ «Amiga Video Toaster» (en inglés). NewTek, Inc. (2010). Consultado el 1 de febrero de 2010.
- 6. ↑ Rabay, Francisco. amiga-hardware.com (ed.): «Newtek: Video Toaster 4000» (en inglés). Consultado el 6 de febrero de 2010.

Bibliografía

- O´brien, James A. (2006). Sistemas de Información Gerencial. México DF.
- Silberschatz, Abraham (2006). Sistemas Operativos. México. ISBN 968-18-6168-X.

Enlaces externos

- Wikipedia.org alberga contenido multimedia sobre Sistema operativo.
- Wikcionario tiene definiciones para sistema operativo.Wikcionario
- Wikiversidad alberga proyectos de aprendizaje sobre Sistema operativo.

From:
https://server-jk.ddns.net/dokuwiki/ - **IES Palomeras-Vallecas Dep. Electronica**

Permanent link:
https://server-jk.ddns.net/dokuwiki/doku.php?id=aula:sistemas_operativos:sistemas_operativos

Last update: **2025/01/22 01:02**