

Sistemas Operativos

Un sistema operativo es un programa que actúa como intermediario entre el usuario y el hardware de un computador y su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas.

Una vez que la computadora ha puesto en marcha su SO, mantiene al menos parte de éste en su memoria en todo momento. Mientras la computadora esté encendida, el SO tiene 4 tareas principales:

- Proporcionar una interfaz de línea de comando o una interfaz gráfica al usuario.
- Administrar los dispositivos de hardware en la computadora.
- Administrar y mantener los sistemas de archivo de disco.
- Apoyar a otros programas. ("llamadas del sistema")

Clasificación:

Sistemas Operativos de multiprogramación (o Sistemas Operativos de multitarea).

Soportan la ejecución de dos o más trabajos activos al mismo tiempo.

Sistemas Operativos de tiempo compartido.

Permiten la simulación de que el sistema y sus recursos son todos para cada usuarios. El usuario hace una petición a la computadora, esta la procesa tan pronto como le es posible, y la respuesta aparecerá en la terminal del usuario.

Sistemas Operativos distribuidos.

Permiten distribuir trabajos, tareas o procesos, entre un conjunto de procesadores. Puede ser que este conjunto de procesadores esté en un equipo o en diferentes, en este caso es transparente para el usuario.

Existen dos esquemas básicos:

- **Sistema fuertemente acoplado:** comparte la memoria y un reloj global, cuyos tiempos de acceso son similares para todos los procesadores.
- **Sistema débilmente acoplado:** Los procesadores no comparten ni memoria ni reloj, ya que cada uno cuenta con su memoria local.

Entre los diferentes Sistemas Operativos distribuidos que existen tenemos los siguientes: Sprite, Solaris-MC, Mach, Chorus, Spring, Amoeba, Taos, etc.

Características de los Sistemas Operativos distribuidos:

- Colección de sistemas autónomos capaces de comunicación y cooperación mediante interconexiones hardware y software
- Objetivo clave: La transparencia.
- Generalmente proporcionan medios para la compartición global de recursos.

Conceptos de los sistemas distribuidos

Los sistemas distribuidos están basados en las ideas básicas de transparencia, eficiencia, flexibilidad, escalabilidad y fiabilidad. Sin embargo estos aspectos son en parte contrarios, y por lo tanto los sistemas distribuidos han de cumplir en su diseño el compromiso de que todos los puntos anteriores sean solucionados de manera aceptable.

Transparencia

Es la idea de que todo el sistema funcione de forma similar en todos los puntos de la red, independientemente de la posición del usuario.

En un sistema transparente, las diferentes copias de un archivo deben aparecer al usuario como un único archivo.

El que el sistema disponga de varios procesadores debe lograr un mayor rendimiento del sistema, pero el sistema operativo debe controlar que tanto los usuarios como los programadores vean el núcleo del sistema distribuido como un único procesador.

Eficiencia

Para lograr un sistema eficiente hay que distribuir las tareas a los procesadores libres más rápidos en cada momento.

Flexibilidad

Un proyecto en desarrollo como el diseño de un sistema operativo distribuido debe estar abierto a cambios y actualizaciones que mejoren el funcionamiento del sistema.

Núcleo monolítico

Como ejemplo de sistema operativo de núcleo monolítico está UNIX. Estos sistemas tienen un núcleo grande y complejo, que engloba todos los servicios del sistema. Está programado de forma no modular, y tiene un rendimiento mayor que un micro núcleo. Sin embargo, cualquier cambio a realizar en cualquier servicio requiere la parada de todo el sistema y la recompilación del núcleo.

micro núcleo

Se basa en una programación altamente modular, y tiene un tamaño mucho menor que el núcleo monolítico. Como consecuencia, el refinamiento y el control de errores son más rápidos y sencillos. Además, la actualización de los servicios es más sencilla y ágil, ya que sólo es necesaria la recompilación del servicio y no de todo el núcleo. Como contraprestación, el rendimiento se ve afectado negativamente.

En la actualidad la mayoría de sistemas operativos distribuidos en desarrollo tienden a un diseño de micro núcleo. Los núcleos tienden a contener menos errores y a ser más fáciles de implementar y de corregir. El sistema pierde ligeramente en rendimiento, pero a cambio consigue un gran aumento de la flexibilidad.

Escalabilidad

Un sistema operativo distribuido debería funcionar tanto para una docena de ordenadores como varios millares. Igualmente, debería no ser determinante el tipo de red utilizada (LAN o WAN) ni las distancias entre los equipos, etc.

Fiabilidad

Una de las ventajas claras que nos ofrece la idea de sistema distribuido es que el funcionamiento de todo el sistema no debe estar ligado a ciertas máquinas de la red, sino que cualquier equipo pueda suplir a otro en caso de que uno se estropee o falle.

La forma más evidente de lograr la fiabilidad de todo el sistema está en la redundancia. La información no debe estar almacenada en un solo servidor de archivos, sino en por lo menos dos máquinas. Mediante la redundancia de los principales archivos o de todos evitamos el caso de que el fallo de un servidor bloquee todo el sistema, al tener una copia idéntica de los archivos en otro equipo.

Otro tipo de redundancia más compleja se refiere a los procesos. Las tareas críticas podrían enviarse a varios procesadores independientes, de forma que el primer procesador realizaría la tarea normalmente, pero ésta pasaría a ejecutarse en otro procesador si el primero hubiera fallado.

Comunicación

La comunicación entre procesos en sistemas con un único procesador se lleva a cabo mediante el uso de memoria compartida entre los procesos. En los sistemas distribuidos, al no haber conexión física entre las distintas memorias de los equipos, la comunicación se realiza mediante la transferencia de mensajes.

Para el envío de mensajes se usa el estándar OSI

Sistemas Operativos de red.

A un Sistema Operativo de Red se le conoce como NOS. Mantiene a dos o más computadoras unidas a través de algún medio de comunicación (físico o no), con el objetivo primordial de poder compartir los diferentes recursos y la información del sistema.

Otra definición es la siguiente; es un software que rige y administra los recursos, archivos, periféricos, usuarios, etc., en una red y lleva el control de seguridad de los mismos.

Un NOS maneja los servicios necesarios para asegurar que el usuario final esté libre de error al acceder a la red. Un NOS normalmente provee una interfaz de usuario que es para reducir la complejidad y conflictos al momento de usar la red.

Dentro del contexto del Sistema Operativo de Red, se pueden escribir aplicaciones tales como un sistema de correo electrónico.

La principal función de un sistema operativo de red es ofrecer un mecanismo para transferir archivos de una máquina a otra. En este entorno, cada instalación mantiene su propio sistema de archivos local y si un usuario de la instalación A quiere acceder a un archivo en la instalación B, hay que copiar explícitamente el archivo de una instalación a otra.

Internet proporciona un mecanismo para estas transferencias, a través del programa protocolo de transferencias de archivos FTP (File Transfer Protocol).

La ubicación del archivo no es transparente para el usuario; tiene que saber exactamente donde esta cada archivo. Además los archivos no se comparten realmente, porque un usuario solo puede copiar un archivo de una instalación a otra. Por lo tanto pueden existir varias copias del mismo archivo, lo que representa un desperdicio de espacio. Así mismo, si se modifican, estas copias no serán consistentes.

El primer Sistema Operativo de red estaba enfocado a equipos con un procesador Motorola 6800, pasando posteriormente a procesadores Intel como Novell Netware.

Los Sistemas Operativos de red más ampliamente usados son: Novell Netware, Personal Netware, LAN Manager, Windows NT Server, UNIX, LANtastic.

Diferencia entre un S.O. Distribuido, un S.O. de Red

No hay que confundir un Sistema Operativo de Red con un Sistema Operativo Distribuido.

En un **Sistema Operativo de Red** las computadoras están interconectadas por medios de comunicación: software y hardware. Los usuarios saben de la existencia de varias computadoras y pueden conectarse con máquinas remotas y copiar archivos de una máquina a otra, cada máquina ejecuta su propio sistema operativo local y tiene su propio usuario o grupo de usuarios.

En cambio en los **Sistemas Operativos Distribuidos** existe un software que distribuye las tareas de los usuarios sobre una red de computadoras y para los usuarios es transparente donde realizan sus tareas y guardan su información.

Aparece ante sus usuarios como un sistema tradicional de un solo procesador, aun cuando esté compuesto por varios procesadores.

Además son sistemas autónomos capaces de comunicarse y cooperar entre sí para resolver tareas globales. Es indispensable el uso de redes para intercambiar datos. Además de los servicios típicos de un Sistema Operativo, un Sistema Distribuido debe gestionar la distribución de tareas entre los diferentes nodos conectados. También, debe proporcionar los mecanismos necesarios para compartir globalmente los recursos del sistema.

Seguridad en los Sistemas Operativos

Dependiendo de los mecanismos utilizados y de su grado de efectividad, se puede hablar de sistemas seguros e inseguros.

En la actualidad son muchas las violaciones que se producen en los sistemas informáticos, en general por acceso de personas no autorizadas que obtienen información confidencial pudiendo incluso manipularla. En ocasiones este tipo de incidencias resulta grave por la naturaleza de los datos; por ejemplo si se trata de datos bancarios, datos oficiales que puedan afectar a la seguridad de los estados, etc.

Software mal intencionado que se produce por diversas causas, es decir pequeños programas que poseen gran facilidad para reproducirse y ejecutarse, con efectos destructivos: virus informáticos.

Para esto, se analizan cuestiones de seguridad desde dos perspectivas diferentes: la **seguridad externa** y la **seguridad interna**.

Seguridad Externa

Engloba a todos los mecanismos dirigidos a asegurar el sistema informático sin que el propio sistema intervenga.

La seguridad externa puede dividirse en dos grandes grupos:

- **Seguridad física.** Engloba aquellos mecanismos que impiden a los agentes físicos la destrucción de la información existente en el sistema; entre ellos podemos citar el fuego, el humo, inundaciones descargas eléctricas, campos magnéticos, acceso físico de personas con no muy buena intención, entre otros.
- **Seguridad de administración.** Engloba los mecanismos más usuales para impedir el acceso lógico de personas físicas al sistema.

Seguridad Interna

Engloba a todos los mecanismos dirigidos a asegurar el sistema informático, siendo el propio sistema el que controla dichos mecanismos.

Estructura de un sistema operativo

Cuatro estructuras distintas que ya han sido probadas, con el fin de tener una idea más extensa de cómo esta estructurado el sistema operativo.

Estructura modular.

También llamados sistemas monolíticos. La estructura consiste en que no existe estructura alguna. El sistema operativo se escribe como una colección de procedimientos, cada uno de los cuales puede llamar a los demás cada vez que así lo requiera.

Estructura por microkernel.

Las funciones centrales de un SO son controladas por el núcleo (kernel) mientras que la interfaz del usuario es controlada por el entorno (Shell).

Por ejemplo, las funciones de bajo nivel del SO y las funciones de interpretación de comandos están separadas, de tal forma que puedes mantener el kernel DOS corriendo, pero utilizar una interfaz de usuario diferente. Esto es exactamente lo que sucede cuando cargas Microsoft Windows, el cual toma el lugar del Shell, reemplazando la interfaz de línea de comandos con una interfaz gráfica del usuario. Existen muchos "shells" diferentes en el mercado, ejemplo: NDOS (Norton DOS), XTG, PCTOOLS, o inclusive el mismo SO MS-DOS a partir de la versión 5.0 incluyó un Shell llamado DOS SHELL.

Estructura por anillos concéntricos (capas).

El sistema por "capas" consiste en organizar el sistema operativo como una jerarquía de capas, cada una construida sobre la inmediata inferior. El primer sistema construido de esta manera fue el sistema THE (Technische Hogeschool Eindhoven), desarrollado en Holanda por E. W. Dijkstra (1968) y sus estudiantes.

- La capa 0 trabaja con la asignación del procesador
- La capa 1 realizaba la administración de la memoria.
- La capa 2 se encargaba de la comunicación entre cada proceso y la consola del operador.
- La capa 3 controla los dispositivos de E/S
- La capa 4 es donde estaban los programas del usuario
- La capa 5 contenía al proceso operador del sistema

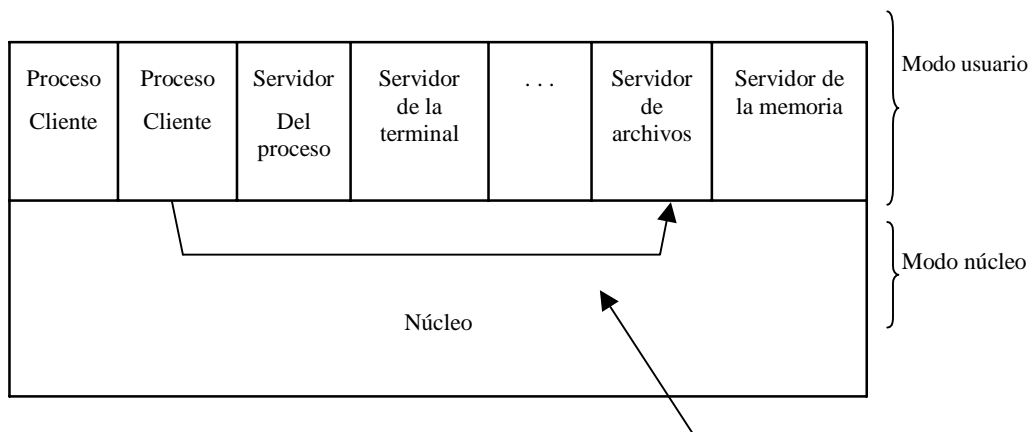
Estructura cliente – servidor

Una tendencia de los sistemas operativos modernos es mover el código a capas superiores y eliminar la mayor parte posible del sistema operativo para mantener un núcleo mínimo.

El punto de vista usual es implantar la mayoría de las funciones del sistema operativo en los procesos del usuario.

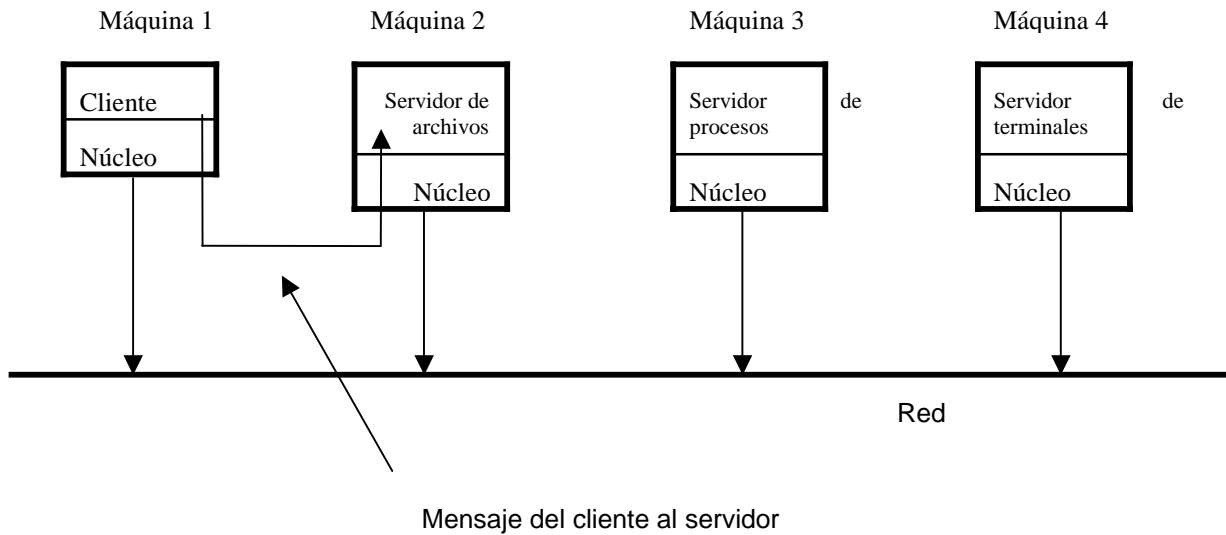
Para solicitar un servicio, como la lectura de un bloque de cierto archivo, un proceso del usuario (denominado proceso cliente) envía la solicitud a un proceso servidor, que realiza entonces el trabajo y regresa la respuesta.

En este modelo, lo único que hace el núcleo es controlar la comunicación entre los clientes y los servidores.



Los clientes obtienen el servicio al enviar mensajes a los procesos servidores

Si un cliente se comunica con un servidor mediante mensajes, el cliente no necesita saber si el mensaje se maneja en forma local, en su máquina, o si se envía por medio de una red a un servidor en una máquina remota. En lo que respecta al cliente, lo mismo ocurre en ambos casos: se envió una solicitud y se recibió una respuesta.



UNIX:

Es un sistema operativo multiusuario que incorpora multitarea. El sistema operativo UNIX esta escrito en lenguaje C. Está disponible en varias formas, entre las que se cuenta AIX, una versión de UNIX adaptada por IBM (para su uso en estaciones de trabajo basadas en RISC), A/ux (versión gráfica para equipos Apple Macintosh) y Mach (un sistema operativo reescrito, pero esencialmente compatible con UNIX, para las computadoras NeXT).

El UNIX y sus clones permiten múltiples tareas y múltiples usuarios. Su sistema de archivos proporciona un método sencillo de organizar archivos y permite la protección de archivos. Sin embargo, las instrucciones del UNIX no son intuitivas.

Orígenes

En el año '71, Ritchie y Kernigham crean y utilizan el lenguaje C en un PDP-11 (algo así como un AT), lenguaje nacido para la programación de sistemas. Dos años después en 1973, Ritchie y Thompson re-escriben su engendro en lenguaje C, pero esta vez desarrollan un sistema multiusuario. UNIX había nacido. El sistema, nacido del trabajo y la ilusión de sólo dos hombres, demostró ser algo tan bueno que ese mismo año Bell Labs contaba con 25 instalaciones funcionando con UNIX.

1984 marca un nuevo hito en la historia de UNIX ya que SUN desarrolla los conceptos de RPC (Remote Procedure Call) y NFS (Network File System) y los incorpora a SunOS.

NFS: Network File System

Es uno de los más conocidos y aceptado como sistema operativo de red. Fue un desarrollo de Sun Microsystems, soportado también por distintos fabricantes:

- Surgió para UNIX pero se amplió a otros S. O. (ej.: MS - DOS).
- Soporta sistemas heterogéneos, por ej.: clientes de MS - DOS que hagan uso de servidores UNIX.
- Los equipos pueden ser también de hardware heterogéneo. Los aspectos más interesantes son los relacionados con:
 - La arquitectura.
 - El protocolo.
 - La implantación.

La Arquitectura de NFS

La idea fundamental es permitir que una colección arbitraria de clientes y servidores compartan un sistema de archivos común.

Generalmente todos los clientes y servidores están en la misma LAN, pero esto no es necesario; por ello se puede ejecutar NFS en una WAN ("red de área amplia").

NFS permite que cada máquina sea un cliente y un servidor al mismo tiempo.

Los clientes tienen acceso a los directorios exportados mediante el montaje:

- Cuando un cliente monta un directorio (remoto), este se convierte en parte de su jerarquía de directorios. Un cliente sin disco puede montar un archivo remoto en su directorio raíz; esto produce un sistema de archivos soportado en su totalidad en un servidor remoto. Las estaciones de trabajo que no poseen discos locales pueden montar directorios remotos.

Si dos o más clientes montan el mismo directorio al mismo tiempo:

- Se pueden comunicar al compartir archivos en sus directorios comunes.
- No hay que hacer nada especial para lograr compartir los archivos.

Los archivos compartidos figuran en la jerarquía de directorios de varias máquinas y se los puede leer o escribir de la manera usual.

Protocolos de NFS

Uno de los objetivos de NFS es:

- Soportar un sistema heterogéneo en donde los clientes y servidores podrían ejecutar distintos S. O. en hardware diverso, por ello es esencial que la interfaz entre los clientes y los servidores esté bien definida. NFS logra este objetivo definiendo dos “protocolos cliente - servidor”:

Un “protocolo” es un conjunto de:

- Solicitudes que envían los clientes a los servidores.
- Respuestas que envían los servidores de regreso a los clientes.

NFS utiliza el esquema de protección de UNIX, con los bits “rwx” para el propietario, grupo y otros.

Resumiendo:

- NFS solo trata el sistema de archivos.
- NFS no hace referencia a otros aspectos, como la ejecución de un proceso.
- NFS se ha difundido ampliamente, a pesar de todo.

NFS es un ejemplo de software débilmente acoplado en hardware débilmente acoplado

- Cada computadora puede ejecutar su propio S. O.
- Solo se dispone de un sistema compartido de archivos.
- El tráfico cliente - servidor debe obedecer los protocolos NFS

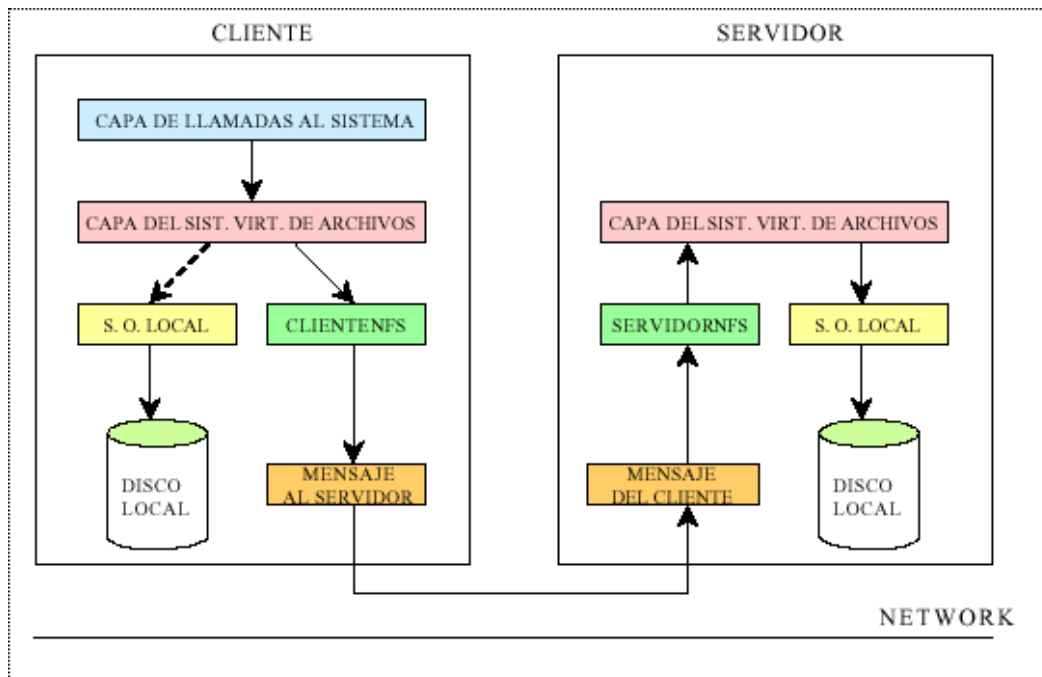


Figura: Estructura de capas de NFS.

Panorama actual: familias de UNIX

Cada fabricante ha ido desarrollando sus estándares: AT&T SVID, el interfaz SYSTEM V; HP, UP-UX (tipo SYSTEM V); DEC, ULTRIX (tipo 4.2BSD); Microsoft, XENIX; e IBM, AIX. Como se puede ver, en líneas generales, existen dos tipos de UNIX: tipo BSD y tipo SYSTEM V. También existen implementaciones de tipo académico, como MINIX, desarrollada por Tanenbaum con afán didáctico en 1983; o XINU, desarrollada por Comer en 1984. La razón de los distintos nombres que recibe el sistema es que UNIX es una marca registrada de AT&T. Así, cada implementación recibe el suyo propio.

Se pueden encontrar muchos otros UNIX: SCO, NetBSD, FreeBSD, LINUX, GNU (...) y cada vez más para el mundo PC.

Minix

Cuando Unix era joven (versión 6), el código fuente se encontraba en todas partes, con autorización de AT&T, y se estudiaba frecuentemente. John Lions, de la Universidad de New South Wales en Australia, llegó a escribir un pequeño folleto que describía su operación, línea por línea. Este folleto se utilizó (con permiso de AT&T) como libro de texto en muchos cursos universitarios sobre sistemas operativos.

Cuando AT&T entregó la versión 7, empezó a comprender que Unix era un valioso producto comercial, así que emitió la versión 7 con una licencia que prohibía el estudio del código fuente en cursos con el objeto de evitar poner en peligro su condición como secreto comercial. Muchas universidades protestaron simplemente descartando el estudio de Unix y enseñando sólo teoría.

Por desgracia, el solo enseñar teoría dejaba al estudiante con una visión desproporcionada de lo que en realidad es un sistema operativo. Los temas teóricos que por lo general se comprenden con lujo de detalle en cursos y libros de sistemas operativos, como los algoritmos de planificación, en la práctica no son realmente tan importantes. Los temas que en realidad son importantes, como la E/S y los sistemas de archivo, por lo general se desprecian porque existe muy poca teoría acerca de ellos.

Para remediar esta situación, el conocido y respetado computólogo, Andrew Tannebaum, decidió escribir un nuevo sistema operativo de la nada que sería compatible con Unix desde el punto de vista del usuario, pero completamente diferente en el interior. Sin siquiera utilizar una sola línea de código de AT&T, este sistema evadía las restricciones de la prohibición, de modo que podía utilizarse para dar una clase o para el estudio individual. En esta forma, los estudiantes podían analizar minuciosamente un sistema operativo real para ver lo que hay en su interior, tal y como los estudiantes de biología diseccionan ranas. El nombre MINIX surgió de mini-Unix porque era lo suficientemente pequeño, para que alguien que no fuera un maestro en sistemas operativos pudiese entender la forma en que trabajaba.

Además de la ventaja de eliminar los problemas legales, MINIX tenía otra ventaja sobre Unix. Aquél se escribió una década después de Unix y se estructuró en forma más modular. El sistema de archivos de MINIX, por ejemplo, no era parte del sistema operativo en absoluto, y corría como un programa de usuario. Otra diferencia es que Unix se diseñó para ser eficiente; MINIX se diseñó para ser legible (en vista de que se puede hablar de que cualquier programa de 12649 líneas es legible). El código de MINIX, por ejemplo, tenía más de 3000 comentarios en él.

Al igual que Unix, MINIX se escribió en lenguaje de programación C y tenía como objetivo ser fácil de portar a diversas computadoras. La implementación inicial se hizo en la IBM PC, puesto que esta computadora tenía un uso extenso. Apegándose a la filosofía de "lo pequeño es bello", MINIX no requería de un disco duro para correr, con lo cual se ajustaba a los presupuestos de muchos estudiantes. MINIX al estar disponible para cualquier estudiante de Ciencias de la Computación del mundo, pronto creó una legión de seguidores, incluyendo sus propios grupos de noticias de USENET.

LINUX

Introducción

El padre de Linux es Linus Torvalds, un programador finlandés de 21 años que inicialmente no tenía más pretensión que 'divertirse' creando un sistema operativo para su uso personal. Torvalds colocó Linux en Internet para que cualquiera lo bajara gratis, en 1991, y desde entonces participan en su desarrollo cientos de voluntarios.

En abril de 1991, Linus Torvalds comenzó a crear un programa que varios meses después se convertiría en Linux, un sistema operativo Unix para PC. Linux nació como un pasatiempo de Torvalds, que en esa época tenía 21 años y estudiaba segundo año de ciencias de los computadores en la Universidad de Helsinki (Finlandia); su intención inicial no era crear un sistema operativo, sino experimentar creando software para manejar ciertas funciones básicas del PC que había comprado cuatro meses antes (un sistema operativo es el programa que controla el funcionamiento de un computador). La inquietud surgió porque el sistema operativo que usaba en su PC, llamado Minix, era una versión de Unix limitada, y él necesitaba algo mejor; Torvalds no usaba DOS o Windows porque le parecían sistemas pobres y poco confiables.

A mediados del año, el programa estaba avanzado y Torvalds comenzó a pensar seriamente en que podría crear el kernel de un nuevo sistema operativo, similar a Minix pero mejor. Torvalds no tenía grandes pretensiones; él dice que no pensó en crear Linux para que fuera un sistema operativo profesional, sino que lo diseñó para su uso personal. Sin embargo, poco a poco su pasatiempo se fue convirtiendo en un proyecto más serio.

El 5 de octubre, Torvalds publicó en un grupo de noticias sobre Minix, en Internet, un mensaje en el que anunció la creación de Linux, un sistema operativo para PC basados en procesadores Intel 386. El mensaje decía que el sistema (Linux versión 0.02) estaba todavía en desarrollo, pero ya funcionaba, y lo ofrecía gratis a cualquiera que deseara bajarlo. También invitó a los programadores interesados en sistemas operativos a usarlo, y enviarle correcciones y mejoras para incluirlas en la próxima versión. Ese fue un suceso clave en la historia de Linux; gracias a Internet, Linux pronto se convertiría en un fenómeno mundial.

El número de usuarios de Linux comenzó a crecer rápidamente, y no era extraño que tuviera gran acogida. Al igual que Torvalds, muchos estudiantes de sistemas amaban los sistemas operativos Unix por su estabilidad y potencia, pero estos eran inalcanzables porque una versión comercial de Unix costaba en esa época 4.000 o 5.000 dólares, y casi todas funcionaban exclusivamente en estaciones de trabajo de 10.000 o más dólares (no en PC). Linux, en cambio, era un sistema Unix gratuito, y funcionaba en PC basados en procesadores Intel (386, 486, etc.).

A medida que creció el número de usuarios, también aumentaron los programadores voluntarios que se involucraron en el desarrollo de Linux. Torvalds distribuyó Linux bajo un tipo de licencia llamada GPL, que permite a cualquier persona bajar, usar, modificar e incluso vender Linux, sin pagar un peso; la única condición es que los cambios o mejoras que una persona o compañía realicen en Linux también deben ser públicos. Esto generó un fenómeno de colaboración mundial sin precedentes. Programadores de todo el planeta enviaron a Torvalds mejoras para el kernel, reportaron errores y comenzaron a crear controladores de dispositivos para Linux. Se calcula que al final de 1992 Linux tenía aproximadamente 1.000 usuarios.

En marzo se lanzó la primera versión 'completa' del sistema operativo de Torvalds: Linux 1.0. Esta versión ofreció soporte a redes de computadores, e incluyó docenas de utilidades, programas de desarrollo de aplicaciones y otras herramientas.

Se fundó Red Hat Software, una empresa que hoy es el principal distribuidor de Linux. Aunque Linux se puede bajar gratis de Internet, hay docenas de empresas –como Red Hat Software y Caldera– que elaboran sus propias versiones, y las venden en CD-ROM, junto con manuales, soporte técnico y

programas adicionales (esto es lo que se conoce en el mundo de Linux como una distribución). Estas distribuciones cuestan entre 10 y 70 dólares, dependiendo de la empresa.

- El 9 de junio se lanzó la versión 2.0 de Linux. Una de las principales novedades fue el soporte a multiprocesamiento simétrico (el sistema aprovechaba el poder de computadores con más de un procesador). Además, Linux 2.0 no solo trabajaba en PC con procesadores Intel x86 (como el 386, 486 y Pentium), sino también en estaciones de trabajo con procesadores Alpha.

- Linus Torvalds se fue a vivir a Santa Clara (California, Estados Unidos), debido a que fue contratado por una compañía llamada Transmeta (es una empresa de chips, que no está relacionada con Linux). Sin embargo, Torvalds continuó encabezando el equipo de gente que se encarga del desarrollo del kernel de Linux.

- La firma de investigaciones Datapro dijo que Linux era el segundo sistema operativo más popular en los servidores web de Internet, después de Solaris (un sistema Unix de Sun Microsystems).

- En septiembre, Intel Corporation y Netscape anunciaron una inversión de capital en la empresa Red Hat Software. Este hecho fue muy importante para aumentar la credibilidad de Linux, debido a que Intel y Netscape son dos de los líderes de la industria de los computadores.

- En diciembre, Corel Corporation lanzó una versión para Linux de su procesador de palabra WordPerfect 8. El programa se colocó en Internet para que los usuarios lo pudieran probar gratis durante 90 días, y en los primeros seis meses lo bajaron un millón de personas.

- En enero se lanzó la versión 2.2 de Linux, que ofreció un mejor rendimiento y soporte para procesadores Sparc, Motorola 68000, PowerPC y MIPS. Esta versión, al igual que la 2.0, soporta computadores con 8 procesadores, pero el multiprocesamiento es mucho más eficiente en la versión 2.2.

- Actualmente, Linux tiene un futuro prometedor. Se calcula que cuenta con más de diez millones de usuarios. Además, trabaja en los principales tipos de procesadores: Intel x86, Motorola 680x0, MIPS, PowerPC (los que usan los Macintosh), Alpha y Sparc (estos dos últimos son procesadores de 64 bits, más potentes que los chips Intel x86). Incluso, hay una versión que funciona en el computador de mano PalmPilot. De otro lado, varios de los principales fabricantes de computadores, como Dell Computer, Compaq y Hewlett-Packard, venden equipos que traen Linux preinstalado.

Funciones

Las funciones principales de este sistema operativo son:

- Sistema multitarea En Linux es posible ejecutar varios programas a la vez sin necesidad de tener que parar la ejecución de cada aplicación.
- Sistema multiusuario Varios usuarios pueden acceder a las aplicaciones y recursos del sistema Linux al mismo tiempo. Y, por supuesto, cada uno de ellos puede ejecutar varios programas a la vez (multitarea).
- Shells programables Un Shell conecta las ordenes de un usuario con el Kernel de Linux (el núcleo del sistema), y al ser programables se puede modificar para adaptarlo a sus necesidades. Por ejemplo, es muy útil para realizar procesos en segundo plano.
- Independencia de dispositivos Linux admite cualquier tipo de dispositivo (módems, impresoras) gracias a que cada una vez instalado uno nuevo, se añade al Kernel el enlace o controlador necesario con el dispositivo, haciendo que el Kernel y el enlace se fusionen. Linux posee una gran adaptabilidad y no se encuentra limitado como otros sistemas operativos.
- Comunicaciones Linux es el sistema más flexible para poder conectarse a cualquier ordenador del mundo. Internet se creó y desarrollo dentro del mundo de Unix, y por lo tanto Linux tiene las mayores capacidades para navegar, ya que Unix y Linux son sistemas prácticamente idénticos. Con Linux podrá montar un servidor en su propia casa sin tener que pagar las enormes cantidades de dinero que piden otros sistemas.

Familia Windows

Windows 3.x

- Es de 16-bit.
- Las dos versiones mas comunes son: - Windows 3.1 y Windows for Workgroups (Windows 3.11).
- Windows 3.1 es un ambiente multitarea para una PC "DOS-based". No tiene capacidad de Networking.

Windows for Workgroups

- Fue diseñado para proveer posibilidades de Networking basadas en Windows y peer-to-peer.
- Una red Windows for Workgroups provee un modo simple de compartir recursos.
- No está previsto para proveer alto nivel de seguridad o para soportar muchos usuarios.

Windows 95 / 98

- Es un sistema operativo de 32-bit.
- Mantiene la posibilidad de ejecutar programas de DOS (16-bit).
- La ganancia de performance es significativa cuando se ejecutan programas de 32-bit.
- El pre-emptive multitasking lo hace mucho mas confiable que Windows 3.x.
- Windows 95 tiene algunos requerimientos de hardware mayores que Windows 3.x, pero fué diseñado para ser compatible con un amplio rango de equipamiento de hardware existente.

Windows 95 es primariamente un sistema operativo desktop (standalone), pero se pueden conectar muchas computadoras Windows 95 juntas para formar una red peer-to-peer, igual que Windows for Workgroups.

Se puede configurar una máquina Windows 95 para que trabaje como si fuera una Workstation en una red NT Server.

Windows NT

Al igual que Windows 95, Windows NT es un sistema operativo de 32-bit versátil. Sin embargo fué diseñado con otras prioridades en mente.

Windows 95 provee máxima compatibilidad con el hardware y las aplicaciones de software existentes, con una instalación mas fácil, mientras que NT provee máxima performance, con confiabilidad industrial, performance y seguridad.

NT tiene requerimientos de hardware mayores que Windows 95. Para óptima performance, debe correr en un Pentium (o superior), con 32MB de RAM. NT funciona en una PC con 16MB de RAM, pero su performance es significativamente superior con 32MB. También requiere mucho mas disco duro que Windows 95.

La interface del usuario de NT 4.0 es bastante similar a la de Windows 95. Las interfaces de versiones previas de NT fueron similares a Windows 3.x. Esto habilita a usuarios que trabajan en ambientes mezclados entre Windows 95/NT 4.0 para utilizar ambos con un mínimo esfuerzo.

Aparte de la interface del usuario, NT y Windows 95 son un poco diferentes de los otros miembros de la familia Windows en un aspecto crucial. NT y Windows 95 son verdaderos Operating Systems. Windows 3.1 y Windows for Workgroups son Environments - Ellos requieren DOS como sistema operativo.

NT comprende dos productos:

- NT Server
- NT Workstation

NT Workstation incluye muchas de las posibilidades de Windows 95, a un nivel mas potente. Puede ser utilizado como:

- Un potente sistema operativo de 32-bit tipo desktop
- En red dentro de un ambiente peer-to-peer workgroup
- Como una Workstation en un ambiente NT Server

Windows NT es muy portable. Está escrito en portable C e incorpora hardware abstraction layer (HAL). Esto permite que NT corra en diferentes tipos de hardware.

Windows NT puede correr en las siguientes plataformas:

- Intel 80x86
- MIPS R4x00
- DEC ALPHA AXP
- PowerPC

Características de NT

- Tiene varias posibilidades que le otorgan escalabilidad. Puede utilizar grandes cantidades de disco rígido y de RAM. Soporta 16 exabytes (EB) de espacio en disco duro y 4 gigabytes (GB) de RAM. (Un exabyte es 1000 millones de GB.)
- Distinto que Windows 3.x y Windows for Workgroups, NT SI soporta full pre-emptive multitasking para todas las aplicaciones. (Windows 95 soporta pre-emptive multitasking para aplicaciones Win32, pero no para aplicaciones Win16.)
- Soporta mas que un stream de ejecución dentro de un proceso (multithreading). Esto le permite mayor concurrencia y performance.
- Cumple con la norma C2 del "US Department of Defense security Standard".
- Tiene muchas opciones de seguridad, incluyendo las siguientes:
 - mandatory logon
 - discretionary Access
 - memory protection
 - auditing
- Soporta los siguientes file systems:
 - NTFS
 - CDFS
 - FAT (16)

New technology file system (NTFS) es el file system nativo de Windows NT.

NT fué diseñado con el "Networking" en mente, y viene con esta habilidad incluida. Ya tiene todos los utilitarios necesarios incluidos, con la habilidad para agregar drivers de red y stacks de protocolos.

NT 4.0 es soportado en las siguientes redes:

- Banyan VINES
- DEC Pathworks
- IBM LAN Server
- Microsoft LAN Manager
- Microsoft Windows for Workgroups
- Novell NetWare
- AppleTalk
- TCP/IP
- NetBEUI.
- IPX/SPX.

También es soportado, con software adicional, en las redes SNA de IBM.

NT soporta clientes de muchas otras redes incluyendo:

- MS-DOS
- Macintosh

- OS/2
- Novell NetWare

Agregar soporte para otros clientes de red, es simplemente cargar la capa de software apropiado en el NT.

NT Workstation otorga cantidad ilimitada de conexiones peer-to-peer. Esto significa que se puede conectar a tantos nodos de red como se requiera. La conectividad Peer-to-peer involucra la interacción de dos nodos, actuando cada uno de ellos como server o cliente según se requiera.

NT Workstation es efectivamente una versión diminuida del NT Server y se desempeñará mas efectivamente que el NT Server en el rol de Workstation.

NT Server es escalable, y soporta hasta cuatro procesadores en configuración SMP, pero hay equipos OEM (original equipment manufacturers) que pueden soportar hasta 32 procesadores.

Arquitectura

Windows NT presenta una arquitectura del tipo cliente-servidor.

Uno de los objetivos fundamentales de diseño fue el tener un núcleo tan pequeño como fuera posible, en el que estuvieran integrados módulos que dieran respuesta a aquellas llamadas al sistema que necesariamente se tuvieran que ejecutar en modo privilegiado (también llamado modo kernel, modo núcleo y modo supervisor). El resto de las llamadas se expulsarían del núcleo hacia otras entidades que se ejecutarían en modo no privilegiado (modo usuario), y de esta manera el núcleo resultaría una base compacta, robusta y estable. Por eso se dice que Windows NT es un sistema operativo **basado en micro-kernel**.

En el diseño de Windows NT han confluído aportaciones de tres modelos: el modelo cliente-servidor, el modelo de objetos, y el modelo de multiprocesamiento simétrico.

Modelo cliente-servidor.

En la teoría de este modelo se establece un kernel que básicamente se encarga de recibir peticiones de procesos clientes y pasárselas a otros procesos servidores, ambos clientes y servidores ejecutándose en modo usuario. Windows NT pone el modelo en práctica pero no contempla el núcleo como un mero transportador de mensajes, sino que introduce en él aquellos servicios que sólo pueden ser ejecutados en modo kernel. El resto de servicios los asciende hacia subsistemas servidores que se ejecutan en modo usuario, independientes entre sí, y que por tanto pueden repartirse entre máquinas distintas, dando así soporte a un sistema distribuido (de hecho, el soportar los sistemas distribuidos fue otra de las grandes directivas de diseño de este SO).

Modelo de objetos.

Decir que no implementa puramente la teoría de este modelo, sino que más bien lo que hace es simplemente contemplar los recursos (tanto internos como externos) como objetos. Brevemente, señalar que todo objeto ha de poseer identidad propia (es único y distinguible de todos los demás), y una serie de atributos (variables) y métodos (funciones) que modifican sus atributos. Los objetos interaccionan entre sí a través del envío de mensajes. No sólo existen en Windows NT objetos software (lógicos), sino que los dispositivos hardware (físicos) también son tratados como objetos (a diferencia de UNIX, que recordemos trataba a los dispositivos como ficheros).

Modelo de multiprocesamiento simétrico.

Un SO multiproceso (o sea, aquel que cuenta con varias CPU y cada una puede estar ejecutando un proceso) puede ser simétrico (SMP) o asimétrico (ASMP). En los sistemas operativos SMP (entre los que se encuentran Windows NT y muchas versiones de UNIX) cualquier CPU puede ejecutar cualquier proceso, ya sea del SO o no, mientras que en los ASMP se elige una CPU para uso exclusivo

del SO y el resto de CPU quedan para ejecutar programas de usuario. Los sistemas SMP son más complejos que los ASMP, contemplan un mejor balance de la carga y son más tolerantes a fallos.

Ventajas de Windows NT.

- Es multitarea y multiusuario.
- Apoya el uso de múltiples procesadores.
- Soporta diferentes arquitecturas.
- Permite el uso de servidores no dedicados.
- Soporta acceso remoto, ofreciendo la detección de intrusos, y mucha seguridad en estas sesiones remotas.
- El sistema está protegido del acceso ilegal a las aplicaciones en las diferentes configuraciones.
- Permite cambiar periódicamente las contraseñas.
- Soporta múltiples protocolos.
- Carga automáticamente manejadores en las estaciones de trabajo.
- Trabaja con impresoras de estaciones remotas.
- Soporta múltiples impresoras y asigna prioridades a las colas de impresión.
- Brinda la posibilidad de asignar diferentes permisos a los diferentes tipos de usuarios.
- Permite realizar diferentes tipos de auditorías, tales como del acceso a archivos, conexión y desconexión, encendido y apagado del sistema, errores del sistema, información de archivos y directorios, etc.

Desventajas de Windows NT.

- No permite criptografía de llave pública ni privada.
- Requiere como mínimo 16 Mb en RAM y un procesador Pentium de 133 MHz o uno superior.
- El usuario no puede limitar la cantidad de espacio en el disco duro.
- No soporta archivos de NFS.

Seguridad.

Windows NT ofrece gran seguridad por medio del acceso por cuentas y contraseñas.

Contiene protecciones para directorios, archivos, y periféricos, es decir que a todo esto se accede con una contraseña para poder utilizarlos.

Windows NT Server ofrece asimismo un modelo lógico de administración que permite administrar de un modo eficaz una red de gran tamaño. Cada usuario sólo necesita disponer de una única cuenta, que se almacena de modo centralizado. Esta única cuenta puede proporcionar al usuario el acceso a cualquier recurso de la red, independientemente del lugar donde se encuentre.

La Conexión con Internet.

Windows NT también hace uso del FTP que es nativo del WEB.

Microsoft adopta finalmente el estilo de UNIX referente a los dominios y lo implanta como una norma.

Dominios.

La administración de una red local bajo Windows NT se basa en los dominios y relaciones de confianza.

La unidad básica de la administración centralizada y la seguridad en Windows NT Server es el dominio. Un dominio es un grupo de servidores que ejecutan Windows NT Server y que, en cierto modo, funcionan como un único sistema.

Todos los servidores con Windows NT Server de un dominio utilizan el mismo conjunto de cuentas de usuario, por lo que sólo es necesario escribir una vez la información de una cuenta de usuario para que todos los servidores del dominio reconozcan dicha cuenta.

Sistemas operativos a futuro

- Los sistemas operativos deben ser extremadamente robustos.
- Ninguna acción del usuario debe hacer que el sistema se comporte de una manera extraña o ilógica.
- Debe ser imposible que cualquier programa interrumpa el funcionamiento de otros programas, o de la maquina en su totalidad. Es decir sin importar las acciones y las decisiones del usuario y del programador, la computadora nunca debe "colgarse". Además, cualquier acontecimiento inesperado se debe manejar de una manera practica. Los problemas que el sistema operativo no puede solucionar se deben señalar al usuario de una manera concisa y clara (tales como un cuadro de dialogo), y se debe explicar de una manera no técnica utilizando un lenguaje dependiendo del modulo de programación.
- El sistema debe enumerar claramente las opciones de las cuales el usuario debe elegir.

El Sistema Operativo del Mañana

Actualmente, hay solamente dos paradigmas fundamentales del sistema operativo:

- El intérprete de la línea de comando (UNIX, DOS)
- La interfaz gráfica (Macintosh OS, OS/2, Windows).

Mientras que el último es claramente más intuitivo y preferido por la mayoría de los usuarios, la industria debe ahora tomar otra medida hacia el sistema operativo ideal. La generación siguiente de sistemas operativos utilizará las nuevas herramientas desarrolladas tales como programación orientada a objetos (OOP), y nueva tecnología de hardware (DRAM's y los microprocesadores densos y baratos), para crear un ambiente que beneficie a los programadores (con modularidad y la abstracción crecientes) así como a los usuarios (proporcionándoles una interfaz interna, constante y gráficamente orientada). Los sistemas operativos futuros también se adaptarán fácilmente a las preferencias cambiantes del usuario y a las tecnologías futuras (tales como sistemas de realidad virtual).

Sistemas operativos e Internet en televisión

En estos años surgió la tendencia de llegar a los hogares de familias que no cuentan con una computadora. Esto es una revolución en lo que se refiere a informática por la razón de que los usuarios de ya no necesitan una computadora para navegar a Internet.

La televisión por computadora ya funciona, y es una de las cuestiones que resultará de máximo interés durante los años de reinado del actual Windows. Lugares como WebTV (<http://www.webtv.com>) ofrecen ya navegación a través de Internet con la tele, y una cantidad de posibilidades de cara al amplio entorno audiovisual mundial. No hace falta tener computadora, y tampoco se necesita un software especial. Lo que sí hace falta es un aparato de conexión que une la línea telefónica con la televisión del usuario.

Los servicios como WebTV multiplican la utilidad de una televisión convencional permitiendo a cualquier persona navegar instantáneamente por la red con un sencillo control remoto, pero además los creadores del concepto anuncian que la fusión de las dos plataformas (receptor de televisión y PC) será una revolución en todos los hogares del mundo.