TEMA

21

Explotación y administración de un Sistema Operativo Multiusuario.

elaborado por
EL EQUIPO DE PROFESORES
DEL CENTRO DOCUMENTACIÓN

El objetivo del sistema operativo es gestionar y administrar eficientemente los recursos hardware del ordenador, permitiendo ejecutar concurrentemente varios programas sin que haya conflictos en el acceso de cada uno de ellos a cada uno de los recursos que necesita y sin que ningún programa monopolice un recurso determinado.

1. DEFINICIÓN DE SISTEMA OPERATIVO

Es un programa o conjunto de programas de control que tiene por objeto facilitar el uso de la computadora y conseguir que esta se ejecute eficientemente.

Es un programa de control ya que se encarga de gestionar y asignar los recursos hardware a los usuarios, entendiendo por recursos hardware: la C.P.U., la memoria principal, discos y otros periféricos.

El S.O. también facilita el uso del ordenador, cada dispositivo de E/S para ser utilizado requiere varias instrucciones maquina que establezcan un diálogo entre la computadora central y el periférico, estas instrucciones dependen considerablemente de las características concretas del periférico y por otra parte estas instrucciones y operaciones son comunes para todos los programas que lo utilizan.

El S.O. con objeto de facilitar el trabajo de los ordenadores contiene rutinas y modelos de gestión de E/S evitando a los programadores y usuarios conocer las instrucciones máquina concretas de cada periférico.

2. TENDENCIA DE LOS SISTEMAS OPERATIVOS

Los sistemas operativos siguen evolucionando. La principal tendencia de los sistemas operativos en cuanto a organización de trabajo es convertirse en sistemas operativos distribuidos.

Los sistemas operativos distribuidos están diseñados para su uso en un grupo de computadoras conectadas pero independientes que comparten recursos. En un sistema operativo distribuido, un proceso puede ejecutarse en cualquier ordenador de la red (normalmente, un ordenador inactivo en ese momento) para aumentar el rendimiento de ese proceso. En los sistemas distribuidos, todas las funciones básicas de un sistema operativo, como mantener los sistemas de archivos, garantizar un comportamiento razonable y recuperar datos en caso de fallos parciales, resultan más complejas.

No hay que confundir un Sistema Operativo de Red con un Sistema Operativo Distribuido. En un Sistema Operativo de Red los ordenadores están interconectados por medios de comunicación: software y hardware. En este tipo de red los usuarios saben donde están ejecutando su trabajo y guardando su información. En cambio en los Sistemas Operativos Distribuidos existe un software que distribuye las tareas de los usuarios sobre una red de ordenadores y para los usuarios es transparente donde realizan sus tareas y guardan su información.

Se ha progresado mucho en el desarrollo de los sistemas operativos. Estos progresos han sido paralelos a la aparición de nuevas tecnologías y de nuevos algoritmos para las tareas de los sistemas operativos.

Actualmente, hay solamente dos paradigmas fundamentales del sistema operativo: el intérprete de la línea de comando (UNIX, DOS), y la interfaz gráfica (Macintosh OS, OS/2, Windows 95, Windows NT, Windows 98).

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 interna, constante y gráficamente orientada interfaz). 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).

3. SISTEMAS OPERATIVOS MULTIUSUARIOS

Estos sistemas operativos son capaces de dar servicio a mas de un usuario a la vez, ya sea por medio de varios terminales conectados al ordenador o por medio de sesiones remotas en una red de comunicaciones.

No importa el número de procesadores en la máquina ni el número de procesos que cada usuario pueda ejecutar simultáneamente.

3.1. EXPLOTACIÓN DE UN SISTEMA OPERATIVO MULTIUSUARIO

El objetivo de la explotación en un sistema operativo multiusuario es obtener la máxima productividad del sistema a través de las posibilidades que ofrecen el conjunto de aplicaciones instaladas sobre el sistema operativo.

Se puede decir que en la actividad de explotación no se tiene en cuenta como está configurado el sistema en su forma global ni quien lo ha hecho, sino que se tiene en cuenta que el sistema funcione de manera eficiente.

Existen muchas actividades a realizar sobre un sistema multiusuario de forma que la productividad sobre este sea máxima, si bien las tareas de explotación se centran en las siguientes:

- Utilización de objetos disponibles en el sistema.
- Utilización del sistema de ficheros.
- Posibilidad de creación de estructuras arboreas de directorios y ficheros.
- Utilización de aplicaciones de carácter general:
  - Correo electrónico.
  - Editor de Textos.
  - Navegador Web.
  - Diseño gráfico.
  - Hoja de Cálculo.
- Posibilidad de configuración y personalización de la cuenta del sistema.
- Acceso a los diferentes dispositivos instalados en el sistema.
- Posibilidad de configurar el entorno de comunicación de nuestra cuenta en el sistema con el exterior.
- Instalación de aplicaciones y programas necesarios para el trabajo sobre la cuenta de mi propiedad.

3.2. ADMINISTRACIÓN DE UN SISTEMA OPERATIVO MULTIUSUARIO

El objetivo de la actividad de administración es proporcionar un sistema donde los usuarios del mismo puedan trabajar de la manera más óptima posible. En este caso la administración se centra en qué se tiene instalar o configurar y como hay que realizarlo para que la productividad del sistema con los recursos disponibles del mismo sea la máxima posible.

En todo sistema multiusuario es necesario tener en cuenta ciertos objetivos para realizar una labor adecuada:

- Analizar los requisitos de cada una de las aplicaciones del sistema. Este análisis solo es necesario realizarlo para aplicaciones ad-hoc disponibles en el sistema.

- Mantener un registro de las instalaciones, componentes y actualizaciones realizadas. Este registro además debe de contener quien realizó la actuación y porqué se realizó.
- Registro de las características y arquitectura de comunicaciones de los equipos que conforman el sistema.
- Mantener documentación del sistema actualizada:
  - Descripción general de los sistemas y recursos del sistema.
  - Volúmenes de información gestionados en la globalidad del sistema.
  - Arquitectura de explotación de cada una de las aplicaciones y utilidades del mismo.

Teniendo presente estos objetivos y la máxima de ofrecer un sistema robusto, fiable y eficiente a todos los usuarios del mismo, es necesario que se actué sobre diferentes elementos u objetos que si bien a bajo nivel son gestionados por el sistema operativo a alto nivel son gestionados por el o los administradores del mismo y se deben centrar en:

- Gestión de cuentas, perfiles y roles de usuarios.
- Gestión de estructura de sistema de ficheros adecuado a las necesidades de los usuarios y de las aplicaciones.
- Gestión de cuotas de disco para cada usuario y aplicación.
- Gestión de dispositivos auxiliares del sistema.
- Gestión de recursos compartidos.
- Gestión de configuración de aplicaciones de propósito general.
- Gestión de seguridad del sistema:
  - Identificar y valorar riesgos.
  - Registro y notificación de supuestos intentos de ataque.
  - Ocultación de información.
  - Cortafuegos (filtrado, proxis).
  - Criptografía.
- Gestión de recuperación de fallos en el sistema.
- Gestión de sistema de backup.
- Gestión de entornos de trabajo.
- Gestión de acceso sobre terminales remotos.
- Gestión eficiente de la memoria.
- Gestión de arquitectura de comunicaciones con el exterior.
- Gestión de Políticas de explotación de las aplicaciones del sistema.
- Gestión de configuración de elementos hardware de uso indirecto.
- Gestión de la monitorización del sistema:
  - Registro automático de eventos.
  - Notificación inmediata de eventos importantes.
- Diagnóstico de problemas y reparación.
- Ayuda a usuarios: resolver problemas, cursos, FAQs.
- Selección de tipos de máquinas y sistemas operativos.

- Selección y dimensionamiento de la red de almacenamiento.
- Selección y dimensionamiento de la red de interconexión.

Hoy en día la mayoría de los sistemas operativos aceptan herramientas de gestión sobre todos los elementos antes descritos si bien, solamente son aquellas mas importantes las que se encuentran disponibles en la distribución software del sistema operativo multiusuario.

De todos estos sistemas operativos, nos centraremos solo en dos, los sistemas operativos monousuario y los sistemas operativos multiusuario.

5. TAREAS DE ADMINISTRACIÓN EN LA EXPLOTACIÓN DE UN SISTEMA OPERATIVO MULTIUSUARIO

En este apartado se describen las tareas que debe tener un administrador en la explotación de un sistema operativo multiusuario.

Objetivos de proceso a corto y largo plazo, acerca de:

- Estudios de viabilidad realizados para el sistema.
- Número de equipos, localización y las características (de los equipos instalados y por instalar y programados) proporcionando entornos de explotación del sistema.
- Fechas de instalación de los equipos, planes de instalación, planes de configuración y planes de explotación.
- Convenios que se tienen con otras instalaciones sobre explotaciones de sistemas operativos, compartiendo la carga total del sistema.
- Configuración de los equipos y capacidades actuales y máximas, determinando de esta forma la carga de explotación posible de cada uno de los equipos.
- Planes de expansión a nivel de explotación.
- Ubicación general de los equipos.
- Políticas de operación.
- Políticas de uso de los equipos.

Una de las tareas más importantes para un administrador es tener muy bien documentado su sistema, para saber en todo momento donde se encuentran los recursos del sistema y así de esta forma sacarlos mayor provecho. Esta documentación estará relacionada con:

- Descripción general de los sistemas instalados que mantengan un umbral de carga de explotación y grandes volúmenes de información.
- Manual de formas de uso de cada equipo en su fase de explotación.
- Manual de procedimientos de los sistemas.
- Descripción genérica.
- Diagramas de entrada, archivos, salida.
- Salidas.
- Fecha de instalación de los sistemas.

6. EXPLOTACIÓN Y ADMINISTRACIÓN DE UN SISTEMA OPERATIVO MULTIUSUARIO

Tener en cuenta

Los sistemas operativos multiusuario, se extenderán en este apartado a través del ejemplo por excelencia, el sistema operativo UNIX. Además de este también se describirá el sistema operativo Windows NT el cual se define como un sistema operativo multitarea pero dependiendo de la tecnologia y configuración (WINDOWS NT SERVER) instalada se podría tener un sistema operativo con las prestaciones muy similares a UNIX, en cuanto al aspecto de SO multiusuario. También se incluye el S.O. Amoeba.

6.1. UNIX

Estructura general de un sistema UNIX

Se puede dividir en varios componentes perfectamente diferenciados:

- Núcleo o Kernel: Comprende un 5-10% del código total.
- Caparazón o Shell: Actúa como intérprete de comandos.
- Programas de utilidad.

Kernel

Es el núcleo del S.O. UNIX. Tiene diversas tareas asignadas, que le sirven como tareas de administración del sistema:

- Planificar, coordinar y gestionar la ejecución de los procesos. Para ello, hace uso de las prioridades asignadas a cada proceso y utiliza algoritmos específicos para repartir el tiempo entre los diversos procesos que compiten por él.
- Dar servicios del sistema, como entrada/salida y gestión de ficheros.

• Manejar las operaciones dependientes de hardware, es decir, realiza las funciones de más bajo nivel de manera que se oculten al usuario.

Shell

Desde el punto de vista del usuario, actúa como un intérprete de comandos. Es un programa que siempre está en ejecución. Esto es uno de los aspectos comunes con el sistema operativo monousuario, aunque el intérprete de comandos de éste sea más sencillo.

El Shell lee las órdenes suministradas, las decodifica y lo comunica al núcleo para realizar la acción especificada.

Prácticamente, todas las órdenes son programas ejecutables que el shell busca en el sistema de ficheros, siguiendo el orden especificado en la variable global PATH.

Explotación y administración de ficheros en UNIX

UNIX emplea un sistema de ficheros jerárquico de directorios-ficheros.

No existe, a nivel de usuario, el concepto de **volumen**, ni de **dispositivo físico**. Es decir, el usuario no sabe en qué disco están los ficheros que está utilizando.

Un fichero (o archivo) es un conjunto de información al que se le da un nombre (nombre del fichero). Existen tres tipos de ficheros en UNIX:

• Ordinarios: Son cadenas de bytes terminadas con <ctrl>D (este código significa fin de fichero). Pueden ser texto, objetos, ejecutables, bibliotecas de módulos,...

• Directorios: Contienen nombres de ficheros y su dirección física. Puede pensarse en ellos como carpetas que contienen ficheros y directorios. Un directorio dentro de otro directorio se denomina subdirectorio.

• Especiales: Asociados a dispositivos entrada/salida. Contienen referencias a los drivers (programas que manejan directamente los dispositivos y que forman parte del núcleo). Pueden ser de tipo "bloque" (apuntan a dispositivos tipo disco) y "carácter" (apuntan a dispositivos como terminales, impresoras, etc). Por convenio, residen en el directorio /dev.

Al elegir los nombres de los ficheros, es conveniente limitarse a utilizar sólo los caracteres que correspondan a letras, números, el carácter subrayado _ y el carácter punto .. Los ficheros cuyo nombre comience por punto permanecen ocultos.

En UNIX existe una jerarquía de directorios estándar, de forma que se simplifica las tareas de administración, esta estructura es:

/	Raiz
/dev	Fichero especiales de dispositivos
/lib	Bibliotecas del sistema
/bin	Ordenes mas empleadas
/etc	Datos y ordenes restringidas al superusuario
/tmp	Ficheros temporales (se borra periodicamente)
/usr	Ordenes, bibliotecas y programas adicionales
/usr/lib	
/usr/bin	
/usr/man	
/usr/...	
/users	Directorios de usuarios

UNIX no mantiene versiones de ficheros, por lo que es necesario prestar especial atención a acciones como borrarlos o modificarlos.

Existen 3 ficheros estándar implementados en UNIX:

- Entrada estándar (stdinput): Teclado (0).
- Salida estándar (stdoutput): Pantalla (1).
- Errores estándar (stderr): Pantalla (2).

La redirección entrada/salida, que se explicará más adelante, permite cambiar estas asignaciones en cualquier momento.

Administración de usuarios en UNIX

UNIX es un sistema operativo multitarea y **multiusuario**, por lo que se deben establecer ciertos mecanismos de tal manera que, simultáneamente, se protejan los datos de un usuario frente a otros y éstos puedan ser compartidos en caso necesario. UNIX posee un mecanismo de **permisos** asociados a cada fichero. Este mecanismo permite que los ficheros y directorios pertenezcan a un usuario en particular. UNIX también permite que los ficheros

sean compartidos entre usuarios y grupos de usuarios. El comportamiento por defecto en la mayoría de los sistemas es que todos los usuarios pueden leer los ficheros de otro usuario, pero no pueden modificarlos o borrarlos.

Los grupos de usuarios se definen normalmente en función del tipo de usuario. Por ejemplo, en una Universidad, los usuarios pueden clasificarse como estudiantes, profesores, invitados, etc.

Cada usuario (perteneciente a un grupo de usuarios) tiene asociado un nombre, una palabra clave o password, un directorio y un proceso de arranque:

- Nombre: Identificación del usuario cuando entra en la máquina (login).
- Clave: Palabra oculta que sólo conoce el usuario.
- UID, GID: Números de identificación de usuario y grupo, respectivamente.
- Directorio: Directorio inicial donde se situará el usuario al entrar en el sistema.
- Proceso: Primer proceso que se arranca una vez dentro del sistema.

Existen diferentes categorías de usuarios en función de sus privilegios (lo que puede y no puede hacer):

- Superusuario o root: Es el administrador del sistema. Tiene todos los privilegios.
- Usuarios normales: El resto de usuarios que pertenecen a distintos grupos, los cuales pueden tener una serie de propiedades comunes.
- Usuarios especiales: Asignados a tareas específicas por el sistema, generalmente de información o manejo de aplicaciones ya instaladas de uso común a usuarios externos o internos. Por ejemplo: mail (se encarga de recoger el correo y repartirlo a los diversos usuarios), lp (se encarga de aceptar trabajos de impresión y mandarlos a la impresora), bin, admin,...

Desde el punto de vista de un usuario, el carácter u significa el propio usuario, g significa el conjunto de usuarios que pertenecen a su mismo grupo, o significa el resto de usuarios. Estos caracteres serán reconocidos por ciertos comandos u órdenes.

Los permisos que llevan asociados todos los ficheros y directorios se clasifican en **lectura** (read, r), **escritura** (write, w) y **ejecución** (execute, x). Estos permisos se pueden asignar al propio **usuario** (u), al **grupo** (g) y al **resto** (o).

El permiso de lectura permite a un usuario leer el contenidos del fichero, o, en el caso de directorios, obtener un listado de su contenido.

El permiso de escritura permite a un usuario escribir y modificar el fichero. Para directorios, permite al usuario crear nuevos ficheros dentro del directorio o borrar los que contiene.

El permiso de ejecución permite a un usuario ejecutar un fichero (debería ser un programa o script –fichero que contiene órdenes para el sistema). En el caso de directorios, permiso de ejecución significa que el usuario puede introducirse en dicho directorio.

Administración de una sesión de trabajo

Como se ha dicho anteriormente, cada usuario tiene asociado un **nombre de usuario** o **cuenta** y una palabra clave o *password*. Para acceder a un sistema y poder trabajar en él, se debe realizar un proceso denominado "login in". El sistema que esté listo para aceptar a un usuario presentará un mensaje como:

login:

a lo que se debe responder con el nombre de usuario asociado. Una vez introducido se pulsa la tecla ENTER o INTRO y el sistema presentará otro mensaje pidiendo la palabra clave:

password:

a lo que responderemos con nuestro password (lo que tecleemos no se presentará en pantalla para evitar que alguien lo pudiera ver), finalizando al pulsar la tecla ENTER o INTRO.
Si cometemos algún error o no tenemos acceso a la **cuenta**, el sistema responderá:

login incorrect

y nos pedirá de nuevo el *login*.

Si todo ha ido bien, el sistema ejecutará una serie de procesos y finalmente aparecerá un *prompt* o petición de órdenes. El prompt es un carácter o conjunto de caracteres que indica que podemos introducir comandos u órdenes; los más típicos son:

$
#
/home/usuario#

Explotación general de sistema

La realización de las tareas más comunes de explotación se suelen llevar a cabo con los siguientes comandos:

date: Fecha y hora del sistema.
cal: Facilita un calendario.
who: Información de quién está en el sistema.
whoami: Información de quién está en este terminal.
pwd: Directorio en el que se está trabajando.
ps: Información de qué es lo que está haciendo el sistema.
df: Información de bloques libres en los discos montados.
man: Manual "on-line".

Administración y explotación con funciones de comunicación

mail: Gestiona el correo electrónico.
write: Nota de entrega inmediata; el mensaje saldrá inmediatamente en el terminal del usuario al que vaya dirigida.
mesg: Inhibe (con opción -n) o permite (con opción - y) la recepción de notas.
sh: invocación a la shell.
rpc: llamadas a procedimientos remotos.
telnet: terminal remoto.
ftp: transferencia de ficheros.

Administración y explotación con funciones de gestión de archivos

cat: Dirige el contenido de ficheros a la salida estándar (normalmente, la pantalla).
cp: Realiza copia de ficheros.
mv: Traslado o cambio de nombre de un fichero.
ln: Establece vínculos entre ficheros.
rm: Borra ficheros. Algunas de las opciones más importantes:
    ls: Lista el contenido de un directorio.
    cd: Cambio de directorio.
    mkdir: Crea un directorio.
    rmdir: Borra un directorio (debe estar vacío).
chown: Cambia el propietario de un fichero (deberá ser nuestro para poderlo hacer).
chgrp: Cambia el grupo de un fichero.

chmod: Cambia los permisos de acceso de un fichero. Actúa sobre el propietario (u), el grupo (g), o el resto (o), añadiendo (+) o quitando (–) los permisos (rwx).
passwd: Para cambiar la palabra clave o *password*. Pide el antiguo y el nuevo dos veces.
lp: Envía peticiones de impresión de ficheros.
cancel: Cancela trabajos pendientes de impresión.
lpstat: Da información sobre las colas de impresión.
shell, permitiendo situar temporalmente a vi "en espera" mientras ejecutas otros comandos. Simplemente sal del caparazón (usando exit) para volver a vi.

6.2. WINDOWS NT

Windows NT es un Sistema Operativo en Red que emplea una estructura cliente/servidor. Frente a las soluciones cliente/servidor tenemos las redes de igual a igual (peer to peer). Con NT tendremos uno o varios servidores que proporcionan recursos y clientes que usan esos recursos. Como clientes se pueden emplear equipos con muchos S.O. diferentes: DOS, Windows 3.1, Windows 95, Windows NT Workstation, UNIX, Macintosh OS y OS/2.

Características generales

Fiabilidad

• **Protección Memoria:** NT proporciona la seguridad de que cuando se ejecuten las aplicaciones del usuario no lo hagan en la zona de memoria que tiene asignada el kernel del sistema. Si una aplicación modificará la zona de memoria ocupada por NT podría dejar colgado el sistema. Por ello NT divide la memoria en anillos. El núcleo del sistema se ejecuta en el anillo 0, mientras que las aplicaciones del usuario se ejecutan en el anillo 3, es decir, las zonas de memoria donde se ejecutan ambos procesos son independientes.

• **Modelo de memoria plana:** NT es un S.O. de 32 bits real. Proporciona un modelo de memoria plana con 32 bits de direcciones, esto permite al S.O. direccionar hasta 4 Gb de memoria.

• **Modelo multitarea preferente:** NT usa la multitarea preferente para garantizar que todas las aplicaciones que se están ejecutando puedan acceder a los recursos del procesador. Es decir, evita que una aplicación monopolice el uso del procesador.

• **Sistema de ficheros transaccional:** NTFS es un sistema de ficheros avanzado y robusto que proporciona una mayor fiabilidad. Es capaz de recuperar una escritura errónea o incompleta. Es similar al TTS de NetWare.

Rendimiento

• **Más rápido:** Al ser un S.O. de 32 bits es mucho más rápido que los S.O. de 16 bits.

• **Multitarea y multiproceso:** Permite ejecutar varias tareas simultáneamente (multitarea) y además soporta varios procesadores en el mismo sistema (multiproceso). La versión Workstation sólo permite dos procesadores mientras el diseño la versión Server permite 32 aunque realmente su implementación sólo admite cuatro procesadores.

• **Multihilo:** Multithreading en inglés (thread hilo). Permite ejecutar distintas partes de una aplicación en paralelo. Cada aplicación consta de un hilo y ese hilo puede tener hilos hijo que se podrían ejecutar en paralelo (sobre todo cuanto contamos con más de un procesador, multiproceso). Cuando se dispone de varios procesadores es un crimen no emplear hilos para ejecutar todas aquellas partes de la aplicación que puedan ser ejecutadas en paralelo.

• **Procesadores RISC:** NT es independiente del hardware, no sólo soporta procesadores INTEL sino que soporta procesadores RISC como son: Power PC, Dec Alpha RISC y MIPSRISC.

Portabilidad

NT, como decíamos, no sólo funciona en plataformas INTEL sino que se puede ejecutar en otros sistemas.

• **Independencia del hardware:** NT tiene un diseño modular que le proporciona independencia del hardware. El único código específico que maneja el hardware reside en el **HAL** (Hardware Abstraction Layer). El HAL opera a bajo nivel traduciendo las operaciones de bajo nivel del S.O. a operaciones que puedan ser entendidas por el hardware específico que se está utilizando. Para dar soporte a un nuevo hardware, se escribe una nuevo HAL para que interactúe con ese hardware y se recompila el S.O.

• **Sistema de archivos configurables:** NT Server soporta múltiples sistemas de archivos: FAT y NTFS. Las versiones anteriores también podían emplear HPFS (High Performance File System) que es el sistema de archivos de OS/2. A partir de la ver-

sión 4 no se reconocen las particiones HPFS. Las particiones HPFS tienen que ser convertidas a FAT o NTFS antes de instalar NT.

Compatibilidad

Un elemento clave para analizar un S.O. es su capacidad para ejecutar las aplicaciones ya existentes. NT se ha diseñado para que sea capaz de ejecutar diferentes aplicaciones e interactúe con diferentes S.O.

- **Diseño de aplicaciones como subsistemas:** NT soporta aplicaciones DOS, Windows 3.x (16 bits), OS/2... NT, como decíamos, tiene un diseño modular lo que le permite ejecutar distintos tipos de aplicaciones. Para ello emplea distintos subsistemas. Para ejecutar un nuevo tipo de aplicación es necesario crear un nuevo subsistema.

- **Subsistema Windows-On-Windows (WOW):** WOW proporciona compatibilidad con las aplicaciones Windows de 16 bits. Ofrece la posibilidad de ejecutar las aplicaciones Windows 3.x en un espacio de memoria compartida o separado.

- **Interfaz de Windows 95:** NT emplea un replica exacta de la interfaz de Windows 95 con algunos objetos menos y otros nuevos.

- **Interoperatividad con NetWare:** NT incluye el protocolo IPX/SPX para clientes NetWare 3.x y NetWare 4.x. Ofrece la capacidad de compartir archivos NetWare, importar cuentas de usuario y login scripts de un servidor NetWare. Se puede migrar de un servidor NetWare a un servidor NT.

- **Interoperatividad con UNIX:** NT se comunica con UNIX a través del protocolo TCP/IP. Incluye aplicaciones de conectividad básicas como FTP, Telnet o Ping.

- **Interoperatividad con Macintosh:** NT soporta el protocolo AppleTalk, que es el protocolo empleado en redes Macintosh. Por ejemplo, NT permite a los sistemas Macintosh la utilización de impresoras conectadas a una red NT.

Seguridad

La seguridad es uno de los aspectos más importante de un S.O. multiusuario.

• **Modelo de seguridad de dominio:** Se trata de un sofisticado sistema de acceso a la red. Dentro de la red existirán unos servidores especiales llamados controladores de dominio que serán los encargados de realizar todo el trabajo de autentificación de usuarios. La información de seguridad se guarda en una base de datos llamada SAM (Security Account Manager).

• **Sistema de archivos NTFS:** Este sistema de archivos complementa la seguridad del sistema, permitiendo a los administradores asignar distintos derechos de acceso a los ficheros y directorios. Además incluye un sistema de control de transacciones similar al TTS de NetWare y la utilidad Hot-Fix.

• **Características de tolerancia a fallos:** NT incluye características de tolerancia a fallos. Tolerancia a fallos significa la capacidad de un sistema para soportar los diferentes errores que se pueda producir durante su funcionamiento. La primera característica importante es el soporte RAID (Redundant Array of Inexpensive Disk) que es similar al disk mirroring y disk duplexing. Otra de las características que incluye es la inclusión de unidades de alimentación ininterrumpidas.

• **Entrada al sistema Ctrl+Alt+Del:** Esta secuencia produce en muchos equipos el reboot del sistema y su consecuente parada y pérdida de datos. Para evitar esto NT ha empleado esta secuencia para entrar al sistema.

Arquitectura del sistema

Comprender cómo funciona Windows NT y cual es su arquitectura es importante para programar aplicaciones para este entorno y recomendable para administrarlo. Vamos a hacer un recorrido por las profundidades de este sistema operativo.

Windows NT presenta una arquitectura del tipo cliente-servidor. Los programas de aplicación son contemplados por el sistema operativo como si fueran clientes a los que hay que servir, y para lo cual viene equipado con distintas entidades servidoras.

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.

Por tanto en un primer acercamiento a la arquitectura distinguimos un núcleo que se ejecuta en modo privilegiado, y se denomina **Executive**, y unos módulos que se ejecutan en modo no privilegiado, llamados **subsistemas protegidos**.

Los programas de usuario (también llamados programas de aplicación) interaccionan con cualquier sistema operativo (SO en adelante) a través de un juego de llamadas al sistema propio de dicho sistema. En el mundo Windows en general, las llamadas al sistema se denominan API (Application Programming Interfaces, interfaces para la programación de aplicaciones). En Windows NT y en Windows 95 se usa una versión del API llamada API Win32.

Subsistemas protegidos

Son una serie de procesos servidores que se ejecutan en modo no privilegiado, al igual que los procesos de usuario, pero que tienen algunas características propias que los hacen distintos.

Se inician al arrancar el s.o. y existen dos tipos: integrales y de entorno.

Un subsistema integral es aquel servidor que ejecuta una función crítica del s.o. (como por ejemplo el que gestiona la seguridad). Un subsistema de entorno da soporte a aplicaciones procedentes de s.o. distintos, adaptándolas para su ejecución bajo Windows NT.
Existen tres de este tipo:

- Win32, que es el principal, y proporciona la interfaz para aplicaciones específicamente construidas para Windows NT.
- POSIX, que soporta aplicaciones UNIX.
- OS/2, que da el entorno a aplicaciones procedentes del s.o. del mismo nombre.

Subsistema WIN32

Es el más importante, ya que atiende no sólo a las aplicaciones nativas de Windows NT, sino que para aquellos programas no Win32, reconoce su tipo y los lanza hacia el subsistema correspondiente. En el caso de que la aplicación sea MS-DOS o Windows de 16 bits (Windows 3.11 e inferiores), lo que hace es crear un nuevo subsistema protegido. Así, la aplicación DOS o Win16 se ejecutaría en el contexto de un proceso llamado VDM (Virtual DOS Machine, máquina virtual DOS), que no es más que un simulador de un ordenador funcionando bajo MS-DOS. Las llamadas al API Win16 serían correspondidas con las homónimas en API Win32. Microsoft llama a esto WOW (Windows On Win32). El subsistema soporta una buena parte del API Win32. Así, se encarga de todo lo relacionado con la interfaz gráfica con

el usuario (GUI), controlando las entradas del usuario y salidas de la aplicación. Por ejemplo, un buen número de funciones de las bibliotecas USER32 y GDI32 son atendidas por Win32, ayudándose del Executive cuando es necesario. El funcionamiento como servidor de Win32 lo veremos un poco más adelante, en el apartado de llamadas a procedimientos locales.

Subsistema POSIX

La norma POSIX (Portable Operating System Interface for UNIX) fue elaborada por IEEE para conseguir la portabilidad de las aplicaciones entre distintos entornos UNIX. La norma se ha implementado no sólo en muchas versiones de UNIX, sino también en otros s.o. como Windows NT, VMS, etc. Se trata de un conjunto de 23 normas, identificadas como IEEE 1003.0 a IEEE 1003.22, o también POSIX.0 a POSIX.22, de las cuales el subsistema POSIX soporta la POSIX.1, que define un conjunto de llamadas al sistema en lenguaje C. El subsistema sirve las llamadas interaccionando con el Executive. Se encarga también de definir aspectos específicos del s.o. UNIX, como pueden ser las relaciones jerárquicas entre procesos padres e hijos (las cuales no existen en el subsistema Win32, por ejemplo, y que por consiguiente no aparecen implementadas directamente en el Executive).

Subsistema OS/2

Igual que el subsistema POSIX proporciona un entorno para aplicaciones UNIX, este subsistema da soporte a las aplicaciones del s.o. OS/2. Proporciona la interfaz gráfica y las llamadas al sistema; las llamadas son servidas con ayuda del Executive.

Subsistema proceso de inicio

El proceso de inicio (Logon Process) recibe las peticiones de conexión por parte de los usuarios. En realidad son dos procesos, cada uno encargándose de un tipo distinto de conexión: el **proceso de inicio local**, que gestiona la conexión de usuarios locales directamente a una máquina Windows NT; y el **proceso de inicio remoto**, el cual gestiona la conexión de usuarios remotos a procesos servidores de NT.

Subsistema de seguridad

Este subsistema interacciona con el proceso de inicio y el llamado **monitor de referencias de seguridad** (del que trataremos en el Executive), de esta forma se construye el modelo de seguridad en Windows NT. El subsistema de seguridad interacciona con el proceso de inicio, atendiendo las peticiones de acceso al sistema. Consta de dos subcomponentes: la **autoridad de seguridad local** y el **administrador de cuentas**.

El primero es el corazón del subsistema de seguridad, en general gestiona la política de seguridad local, así, se encarga de generar los permisos de acceso, de comprobar que el usuario que solicita conexión tiene acceso al sistema, de verificar todos los accesos sobre los objetos (para lo cual se ayuda del monitor de referencias a seguridad) y de controlar la política de auditorías, llevando la cuenta de los mensajes de auditoría generados por el monitor de referencias.

El **administrador de cuentas** mantiene una base de datos con las cuentas de todos los usuarios (login, claves, identificaciones, etc.). Proporciona los servicios de validación de usuarios requeridos por el subcomponente anterior.

El Executive

No debemos confundir el Executive con el núcleo de Windows NT, aunque muchas veces se usan (incorrectamente) como sinónimos. El Executive consta de una serie de componentes software, que se ejecutan en modo privilegiado, uno de los cuales es el núcleo. Dichos componentes son totalmente independientes entre sí, y se comunican a través de interfaces bien definidas. Recordemos que en el diseño se procuró dejar el núcleo tan pequeño como fuera posible y, como veremos, la funcionalidad del núcleo es mínima.

El administrador de objetos

Se encarga de crear, destruir y gestionar todos los objetos del Executive. Tenemos infinidad de objetos: procesos, subprocesos, ficheros, segmentos de memoria compartida, semáforos, mutex, sucesos, etc. Los subsistemas de entorno (Win32, OS/2 y POSIX) también tienen sus propios objetos. Por ejemplo, un objeto ventana es creado (con ayuda del administrador de objetos) y gestionado por el subsistema Win32. La razón de no incluir la gestión de ese objeto en el Executive es que una ventana sólo es innata de las aplicaciones Windows, y no de las aplicaciones UNIX o OS/2. Por tanto, el Executive no se encarga de administrar los objetos relacionados con el entorno de cada s.o. concreto, sino de los objetos comunes a los tres.

El administrador de procesos

Se encarga (en colaboración con el administrador de objetos) de crear, destruir y gestionar los procesos y subprocesos. Una de sus funciones es la de repartir el tiempo de CPU entre los distintos subprocesos. Suministra sólo las relaciones más básicas entre procesos y subprocesos, dejando el resto de las interrelaciones entre ellos a cada subsistema protegido concreto. Por ejemplo, en el entorno POSIX existe una relación filial entre los procesos

que no existe en Win32, de manera que se constituye una jerarquía de procesos. Como esto sólo es específico de ese subsistema, el administrador de objetos no se entromete en ese trabajo y lo deja en manos del subsistema.

El administrador de memoria virtual

Windows NT y UNIX implementan un direccionamiento lineal de 32 bits y memoria virtual paginada bajo demanda. El VMM se encarga de todo lo relacionado con la política de gestión de la memoria. Determina los conjuntos de trabajo de cada proceso, mantiene un conjunto de páginas libres, elige páginas víctima, sube y baja páginas entre la memoria RAM y el archivo de intercambio en disco, etc.

El administrador de entrada/salida

Consta de varios subcomponentes: el **administrador del sistema de ficheros**, el **servidor de red**, el **redirector de red**, los **controladores de dispositivo del sistema** y el **administrador de cachés**.

Buena parte de su trabajo es la gestión de la comunicación entre los distintos controladores de dispositivo, para lo cual implementa una interfaz bien definida que permite el tratamiento de todos los controladores de una manera homogénea, sin preocuparse del funcionamiento específico de cada uno. Trabaja en conjunción con otros componentes del Executive, sobre todo con el VMM. Le proporciona la E/S síncrona y asíncrona, la E/S a archivos asignados en memoria y las caches de los ficheros. El administrador de caches no se limita a gestionar unos cuantos buffers de tamaño fijo para cada fichero abierto, sino que es capaz de estudiar las estadísticas sobre la carga del sistema y variar dinámicamente esos tamaños de acuerdo con la carga. El VMM realiza algo parecido en su trabajo.

El monitor de referencias a seguridad

Este componente da soporte en modo privilegiado al subsistema de seguridad, con el que interacciona. Su misión es actuar de alguna manera como supervisor de accesos, ya que comprueba si un proceso determinado tiene permisos para acceder a un objeto determinado, y monitoriza sus acciones sobre dicho objeto. De esta manera es capaz de generar los mensajes de auditorías. Soporta las validaciones de acceso que realiza el subsistema de seguridad local.