ARQUITECTURA DEL SISTEMA OPERATIVO

MULTITAREA

El término multitarea, se refiere a la capacidad que tiene un Sistema Operativo para compartir el Microprocesador entre distintos programas.

Para referirnos a un programa cargado en memoria y ejecutándose (definición de tarea), se emplea más el término Proceso. De hecho, en Windows 95 desaparece totalmente la palabra tarea, muy utilizada en Windows 3x y sólo se emplea la palabra Proceso, como ocurre en Windows NT, Unix, etc.

El componente que se encarga de administrar los procesos de Windows 95 es el Planificador de Procesos, que además proporciona recursos del sistema a las Aplicaciones. Este Planificador de Procesos de Windows 95 trabaja con dos modelos de multitarea:

Multitarea Prioritaria: en un sistema multitarea con asignación prioritaria, cada thread se ejecuta durante un período de tiempo preestablecido o hasta que haya otro thread con prioridad superior que esté preparado para ejecutarse. La planificación la lleva a cabo el sistema operativo sin la participación de la aplicación, por lo que resulta más difícil para una aplicación o para un thread monopolizar el procesador. Para impedir que threads de diferentes procesos accedan a un recurso que no se puede compartir, la aplicación puede establecer semáforos (indicadores especiales utilizados por la aplicación) para bloquear el recurso hasta que se deje de utilizar.

En Windows 95, las aplicaciones basadas en MS−DOS y las de 32 bits basadas en Windows se ejecutan en multitarea prioritaria.

Multitarea Cooperativa: en este tipo de multitarea, (llamada también sin asignación prioritaria), un thread se ejecuta hasta que renuncia voluntariamente al procesador. La aplicación determina cuándo deja de ejecutarse el thread.

En Windows 95, las aplicaciones de 16 bits basadas en Windows se ejecutan en multitarea cooperativa. El repartidor multitarea con asignación prioritaria trata a todas las aplicaciones de 16 bits basadas en Windows como una única tarea. Esta multitarea híbrida es necesaria para mantener la compatibilidad con las aplicaciones de 16 bits basadas en Windows que esperan poder controlar su propia ejecución.

PROCESOS Y THREADS

Windows 95 tiene múltiples Threads, lo cual significa que en un momento determinado puede ejecutarse más de un thread de ejecución (o thread) en una única tarea o aplicación.

Definición de Proceso: un proceso es esencialmente una aplicación. Cada proceso tiene memoria y acceso a los recursos del sistema. Un Proceso comprende:

- Un programa ejecutable que define los datos y el código iniciales.
- Un espacio de direcciones de memoria en el que se almacenan los datos y el código del proceso.
- Recursos del sistema como archivos y ventanas.
- Al menos un thread para ejecutar el código. Las aplicaciones MS−DOS y las basadas en Windows de 16 bits tienen un solo thread por proceso. Las aplicaciones de 32 bis basadas en Windows pueden tener varios threads por proceso.

Definición de Thread: un thread es una unidad de ejecución. Es el componente real de un proceso que se ejecuta en un momento dado. Se ejecuta en el espacio de direcciones del proceso y utiliza los recursos asignados a dicho proceso.

NOTA: la propiedad de los recursos corresponde al proceso, no a los threads. Los threads utilizan los recursos, pero el proceso conserva la propiedad.

Un thread comprende:

- Un estado del procesador que incluye el puntero de la instrucción actual (registro).
- Una pila para utilizarla cuando se ejecuta en modo de usuario.
- Una pila para utilizarla cuando se ejecuta en modo de núcleo.

En un programa de múltiples threads, el programador tiene la responsabilidad de garantizar que los diferentes threads no interfieran con otros. Esto puede conseguirse utilizando los recursos compartidos de forma que no se entre en conflicto con otro thread que esté utilizando el mismo recurso.

PRIORIDADES DE LOS THREADS

Cada thread tiene una prioridad base. La prioridad determina cuándo se ejecuta un thread en relación con otros threads del sistema. Se otorga el uso del procesador al thread que tenga la mayor prioridad. La prioridad base de los threads de un proceso puede alterarse en dos niveles ascendentes o descendentes.

Hay 32 niveles de prioridad y el planificador puede cambiar la prioridad de los threads de las aplicaciones.

Planificación: es el proceso por el cual se determina qué thread debe utilizar el procesador. Este proceso se basa en una unidad de tiempo predeterminada. En la práctica, el lapso de tiempo real depende de la configuración del sistema.

Hay dos planificadores, el primario y el secundario. El primario examina todos los threads que se están ejecutando y obtiene su número de prioridad. A continuación compara las prioridades y asigna los recursos consecuentemente. Se ejecuta el thread que tenga la mayor prioridad. Si dos o más threads tienen la misma prioridad, se ponen en una pila y el que se encuentra en la parte superior se ejecuta durante el lapso de tiempo predeterminado y luego se desplaza a la parte inferior de la pila; entonces se ejecuta, también durante el lapso predeterminado, el siguiente thread que se encuentra en la parte superior de la pila y, después, se traslada al fondo y así sucesivamente. Cada thread se ejecuta durante el lapso predeterminado y luego se desplaza al final de la pila. Este proceso continúa hasta que no haya más threads con la misma prioridad.

Después de este ciclo de ejecución, el planificador secundario eleva las prioridades de los threads que no se están ejecutando para que los que tengan una prioridad menor al principio alcancen una mayor y puedan ejecutarse. Esto impide que un thread con una prioridad baja pueda quedar bloqueado y, en consecuencia, se impida su ejecución.

El planificador secundario también cambia la prioridad de los threads conforme va pasando el tiempo con el fin de aligerar el funcionamiento global de las aplicaciones. Según el tipo de trabajo que esté realizando el thread, Windows 95 puede ajustar la prioridad del mismo de forma ascendente o descendente a partir de su prioridad base. Por ejemplo:

- Se aumenta la prioridad de los threads que esperan la entrada del usuario (threads del proceso de primer plano). De esta manera el sistema muestra un mayor interés por el usuario.
- Se aumenta la prioridad de los threads que han completado una espera voluntaria.
- Todos los threads reciben periódicamente un aumento de prioridad para impedir que los de menor

prioridad bloqueen recursos compartidos que otros threads con una prioridad mayor necesitan.

- Se rebaja la prioridad de los threads restringidos por la velocidad de cálculo. De esta forma se impide que se bloqueen las operaciones de E/S.

EQUIPOS VIRTUALES

En los equipos virtuales (VM) existen aplicaciones. Un equipo virtual es un entorno creado por el sistema operativo y por el procesador que simula los recursos de todo el equipo (controladores de disco, cronómetros, etc.). El equipo virtual se presenta ante una aplicación como si se tratara de un equipo completo; por ello las aplicaciones tienen acceso virtual al hardware y a todos los demás recursos.

Los equipos virtuales hacen que la programación sea más sencilla. El programador no tiene que preocuparse de hacer un seguimiento del uso del hardware de otras aplicaciones. LA aplicación hace una llamada a un dispositivo virtual. El sistema operativo realiza el seguimiento de las aplicaciones y del hardware, y determina de qué recursos puede disponer cada aplicación.

Equipos virtuales en Windows 95: todas las aplicaciones basadas en Windows 95 funcionan en un equipo virtual (VM). Existen varios VM que pueden utilizarse con Windows 95, que también dispone de un VM del sistema y puede tener varios VM del MS−DOS. El VM del sistema contiene:

- Componentes del sistema base (Núcleo, Usuario, GDI).
- Un espacio de direcciones compartidas por aplicaciones basadas en Windows de 16 bits.
- Un espacio de direcciones separado para cada aplicación basada en Windows de 32 bits.

Aplicaciones basadas en MS−DOS: debido a que las aplicaciones basadas en MS−DOS esperan poder disponer de acceso directo y exclusivo al hardware, cada aplicación individual basada en MS−DOS que se ejecuta en Windows 95 utiliza un VM distinto.

Aplicaciones basadas en Windows de 16 bits: estas aplicaciones comparten un único espacio de direcciones en el VM del sistema. Es posible que una aplicación de 16 bits entre en la RAM de otra aplicación basada en Windows de 16 bits. Esto es necesario para permitir la compatibilidad con las aplicaciones de 16 bits existentes que esperan poder interactuar entre sí compartiendo la memoria de esta forma.

Aplicaciones basadas en Windows de 32 bits: cada una de estas aplicaciones tiene su propio espacio de direcciones en el VM del sistema. Cada dirección de la aplicación de 32 bits está protegida por el hardware. No pueden entrar en la RAM de otras.

MENSAJERÍA INTERNA DE WINDOWS

Windows utiliza un modelo de paso del mensaje para controlar las aplicaciones. Los mensajes se generan siempre que se produce un evento, como ser si el mouse inicia o detiene su movimiento o si se presiona una tecla del teclado, se produce una interrupción. Estas interrupciones se convierten en mensajes. Las aplicaciones Windows también crean mensajes para solicitar al sistema operativo que realice un servicio o que transmita información. Los mensajes se colocan en la cola de mensajes adecuada.

Este procesamiento asíncrono de mensajes significa que cada cola se procesa independientemente. Si una aplicación queda bloqueada (y ya no lee sus mensajes) ello no impide que las demás colas procesen sus mensajes.

Las aplicaciones basadas en Windows de 16 bits comparten una cola de mensajes común. Si una aplicación basada en Windows de 32 bits se queda bloqueada, toda las aplicaciones basadas en Windows de 16 bits que

estén en ejecución tendrán sus mensajes bloqueados hasta que se quite la aplicación colgada.

Debido a que las aplicaciones basadas en MS−DOS no utilizan el diseño de paso de mensajes, no disponen de ninguna cola de mensajes.

FUNCIONES DEL SISTEMA OPERATIVO: BIBLIOTECAS DE VÍNCULOS DINÁMICOS

Windows 95 ha incorporado diversas funciones en el sistema operativo para hacer que el uso de las aplicaciones sea más fácil y rápido. Cuando una aplicación necesita utilizar una de las funciones incorporadas, se envía un mensaje al sistema operativo.

Estas funciones se encuentran en las bibliotecas de vínculos dinámicos (las DLL) respectivas.

Una biblioteca de vínculos dinámicos es una colección de rutinas ejecutables que pueden compartirse. Estas rutinas se pueden cargar y ejecutar durante la ejecución de una aplicación.

Windows se compone principalmente de tres bibliotecas de vínculos dinámicos:

- **Kernel:** KERNEL32.DLL gestiona las funciones del sistema operativo base, como la administración de la memoria, la E/S de archivos, la carga de aplicaciones y la ejecución.
- **User:** USER32.DLL y USER.EXE controlan la entrada y la salida, incluyendo el teclado, el mouse y el controlador de sonido, el cronómetro y los puertos de comunicaciones. Proporcionan la interfaz de usuario de Windows incluidos la colocación y movimiento de ventanas, íconos y cuadros de diálogo.
- **GDI:** GDI32.DLL Y GDI.EXE administran los gráficos y la impresión. Esto incluye la activación y desactivación de pixeles, el dibujo de líneas, el dibujo de gráficos, el dibujo en la pantalla y el dibujo en la impresora.

Windows 95 dispone de DLL adicionales que colocan otras funciones directamente en el sistema operativo. Estas DLL incluyen WinNet, multimedia y OLE.

- **WinNet:** las DLL WinNet permiten el acceso independiente del dispositivo a las funciones de la red.
- **Multimedia:** gestiona la interfaz para los controladores de dispositivos multimedia y las funciones de E/S. Esto incluye aspectos tales como los algoritmos de compresión y descompresión, y la sincronización de audio y video.
- **OLE:** es el sistema de administración de objetos de Microsoft. Incluye los mecanismo para el almacenamiento estructurado, el uso compartido de datos, los documentos compuestos y la manipulación de objetos.

CONTROL DE DISPOSITIVOS DE HARDWARE: CONTROLADORES DEL SISTEMA DE WINDOWS

Un controlador es un conversor de software. Acepta los diversos comandos del sistema operativo y los convierte en comandos específicos para un elemento determinado del hardware. La utilización de controladores permite que las aplicaciones sean independientes del hardware, que puede cambiarse sin que ello afecte a la aplicación.

Necesidad del uso de controladores

Los controladores conectan Windows al hardware y proporcionan independencia del dispositivo. Cuando se crea hardware nuevo, también puede crearse un controlador nuevo; no es necesario cambiar Windows. Algunos de los tipos de controladores utilizados por Windows son:

- Sistema
- Teclado
- Mouse
- Pantalla
- Sonido
- Comunicaciones
- Impresora
- Adaptador de red

Tipos de Controladores

Para conservar la compatibilidad con las aplicaciones existentes, Windows 95 permite utilizar los siguientes controladores:

- Modo real de MS−DOS (*.SYS)
- Windows en modo real (normalmente aparece en SYSTEM.INI, *.DRV)
- Windows en modo protegido (controladores virtuales *.VXD ó *.386)

Si un nuevo controlador no da soporte a un dispositivo, el controlador de MD−DOS del archivo CONFIG.SYS se queda en su lugar y sigue funcionando. Si el proceso de instalación no puede garantizar que el controlador puede eliminarse sin riesgos, el controlador permanece en su lugar y se utiliza el controlador de Windows de 32 bits tras cambiar al modo protegido. Si para el proceso de inicio no se necesita el controlador, éste puede eliminarse del archivo CONFIG.SYS. Sin embargo, los controladores en modo protegido deberán utilizarse siempre que sea posible.

Controladores en modo protegido frente a controladores en modo real: los controladores en modo real son controladores creados para ejecutarse bajo el sistema operativo MD−DOS en modo real. Los controladores en modo real no son tan seguros ni tan sólidos como los controladores en modo protegido.

Los controladores en modo protegido aprovechan la arquitectura del modo protegido de los procesadores 80386 y superiores. Uno de éstos controladores (también denominado controlador de dispositivo virtual o VxD) permite un acceso compartido más rápido al dispositivo sin utilizar los 640 Kb inferiores de la memoria.

Archivos de Controlador

La mayoría de los controladores de Windows se encuentran en el directorio \WINDOWS\SYSTEM y normalmente se llaman *.DRV; *.386 o *.VXD. Los controladores *.DRV son controladores basados en Windows de 16 bits más antiguos, mientras que los controladores *.VXD y *.386 son controladores en modo protegido de 32 bits.

Los controladores *.VXD que siempre se utilizan en una instalación de Windows 95 específica, son combinados por el proceso de instalación en el archivo VMM32.VXD para acelerar la carga. Los nuevos controladores del sistema, instalados tras la instalación de Windows 95, se colocan en la carpeta \WINDOWS\SYSTEM\VMM32 y se agregan a VMM32.VXD la próxima vez que se ejecuta el programa de instalación.

REGISTRO DE CONFIGURACIÓN DE WINDOWS

El registro de configuraciones es el lugar donde el sistema y las aplicaciones almacenan los datos. Es similar a como los archivos *.INI almacenaban los datos en las versiones anteriores de Windows. Entre las ventajas de utilizar el registro de configuraciones se incluyen las posibilidades de asignar varios valores a una clave y poder asociar diferentes tipos de datos a las claves. También se puede tener acceso al registro de

configuraciones a través de una red con fines de diagnósticos y administración utilizando las herramientas remotas del registro.

Estructura del Registro de configuraciones

El registro de configuraciones es una base de datos jerárquica con estructura de árbol que se almacenan en dos archivos. Los archivos reales que se utilizan pueden variar en funciones de la configuración del sistema, pero se dividen en un archivo que contiene configuraciones específicas de la máquina (normalmente SYSTEM.DAT) y un archivo que contiene configuraciones específicas del usuario (normalmente USER.DAT).

El registro de configuraciones tiene estructura de árbol, de forma parecida al registro de versión 3.1 de Windows. A continuación se presentan las seis claves de raíz del registro de configuración de Windows 95.

HKEY_CLASSES_ROOT: es la información acerca de las asignaciones de la asociación de OLE2 y de archivo. Es la que permite que Windows se ejecute o imprima desde una aplicación cuando se selecciona un archivo de datos específicos.

HKEY_USERS: la clave de los usuarios contiene información acerca de todos los usuarios de esta estación de trabajo. Aquí es donde se almacena la información sobre cada usuario. Además, existe una configuración de usuario genérica.

La información contenida se relaciona directamente con las diversas configuraciones predeterminadas para las aplicaciones, esquema de eventos, configuraciones de escritorio, etc.

HKEY_CURRENT_USER: esta clave contiene la configuración específica del usuario para el sistema y las aplicaciones. Se crea un tiempo de ejecución a partir de la información que se carga desde la entrada de ese usuario en la sección del registro de configuración HKEY_USERS a esta clave del registro de configuraciones. Se trata de información acerca de la forma en que el usuario desea configurar su entorno de estación de trabajo. Otra información podría ser las combinaciones de colores, etiquetas, el estado del escritorio, etc., que prefiere el usuario.

HYKEY_LOCAL_MACHINE: la clave de máquina local contiene especificaciones para la estación de trabajo, los controladores y otras configuraciones del sistema es información específica de la máquina acerca del tipo de hardware instalado, las asignaciones de los puertos, la configuración actual de software, etc. Esta información es específica de la máquina y no por usuario.

HKEY_CURRENT_CONFIG: la clave contiene información acerca de la configuración actual de hardware conectado a la estación de trabajo. Esta clave se utiliza principalmente cuando existen varias configuraciones para la estación de trabajo. La información de esta clave se copia a partir de la información de configuración contenida en la clave HKEY_LOCAL_MACHINE.

HKEY_DYN_DATA: esta clave contiene la información de estado dinámico de diversos dispositivos. La información que se encuentra en esta clave se vuelve a generar cada vez que se inicia el sistema. Esta clave se utiliza como parte de la información de medida del rendimiento y también para la configuración Plug & Play. Esta información puede cambiar a medida que se agregan o eliminan dispositivos del sistema. La información que se incluye para cada dispositivo se compone de la clave del hardware asociado, cualquier problema que se haya producido y el estado actual del dispositivo. Esta clave también contiene información sobre la supervisión del sistema que está realizándose con la herramienta monitor del sistema HKEY_DYN_DATA no forma parte de ningún archivo de registro de configuraciones y siempre se crea dinámicamente.

Modificación del registro de configuración

En la mayoría de los casos no es necesario modificar el registro de configuraciones. Las modificaciones al sistema se realizan de forma más correcta mediante la interfaz del usuario adecuada, como el administrador de dispositivos u otras partes del panel de control.

Si es absolutamente necesario modificar el registro de configuraciones, existe una herramienta que puede utilizar.

El editor de registros se encuentra en la carpeta de Windows. Es una aplicación que le permite ver y modificar la configuración actual del registro.

- Para ver la configuración con el editor de registro:

- Desde la carpeta de Windows inicie el editor de registros (se llama REGEDIT.EXE).
- Seleccione la clave que contiene los datos que desea ver resaltando dicha clave.

Si una clave contiene subclaves adicionales, aparecerá un signo + junto a dicha clave. Para ver las subclaves haga click en el signo − para ver el valor de una clave, haga click en el signo + que se encuentra junto a la clave.

El editor de registros también contiene una función de búsqueda. En el menú editar, seleccione buscar e introduzca la cadena que desea encontrar.

ADMINISTRACIÓN DE MEMORIA

Memoria Virtual

Windows 95 virtualiza de la misma forma que virtualiza las máquinas y los controladores. Esta virtualización de la memoria permite que las aplicaciones se comporten como si cada uno tuviera su propia RAM física.

La memoria virtual es la forma en que un programa ve la memoria.

A cada aplicación se le asigna un espacio de direcciones virtuales, que es el conjunto de direcciones que la aplicación puede utilizar. A cada aplicación de 32 bits y a cada aplicación basada en MS_DOS se les asigna su propio espacio de direcciones virtuales. Las aplicaciones acceden a la memoria a través de la direcciones virtuales, que el administrador de la memoria de Windows 95 asigna a direcciones físicas. Estas direcciones físicas pueden señalar ubicaciones de la RAM o el disco duro.

En Windows 95, el sistema operativo puede asignar más memoria de la que está físicamente disponible de la estación de trabajo. Windows 95 ofrece memoria virtual utilizando las posibilidades de paginación en función de la demanda del procesador 80386 de Intel o superior. El código del programa y los datos que se encuentran en la memoria física pueden trasladarse a un archivo de intercambio creado para ese fin en el disco duro.

El archivo de intercambio cambia de tamaño según sean las necesidades del sistema. Si el espacio en la unidad de disco duro empieza a escasear, se disminuye el tamaño del archivo de intercambio. Si hay espacio disponible en el disco duro y se necesita más memoria, se aumenta el tamaño del archivo de intercambio.

Cada espacio de direcciones está dividido en 1.048.576 (220) páginas y cada página tiene un tamaño de 4 Kb. La memoria física se asigna página a página.

Una página puede indicar memoria que se encuentra en un lugar de la RAM o en un archivo de intercambio de un disco, o bien puede estar marcado como no utilizada. En este último caso, la dirección existe, pero no se le ha asignado memoria física.

El administrador de memoria virtual asigna las direcciones virtuales del espacio de direcciones del proceso a las páginas físicas de la memoria o del archivo de intercambio del equipo, con lo cual oculta a la aplicación de organización física de la memoria. De esta forma se asegura que los threads puedan tener acceso a la memoria del proceso al que pertenecen a medida que la necesiten, pero no a la memoria de otros procesos.

El administrador de memoria virtual de Windows controla todo el proceso de intercambio que tiene lugar en el disco. Lleva a cabo la paginación y mantiene una lista de las páginas de 4 Kb que se encuentran en ese momento en la memoria física accediendo a una tabla de páginas. Dicha tabla indica a Windows qué páginas se han trasladado al disco, cuáles pertenecen a cada proceso, etc.

Direcciones Virtuales

Windows 95 asigna a cada espacio de memoria virtual 4 GB de direcciones de memoria. Se trata sólo de espacio de direcciones, no se necesitan 4 Gb de memoria física. Las direcciones de memoria se asignan de a siguiente forma:

- **0 − 1 MB:** es una memoria virtual de MS−DOS, estas direcciones las utilizan las aplicaciones basadas en MS−DOS. Si no se trata de una memoria virtual de MS−DOS, las aplicaciones no utilizan las direcciones no que éstas quedan disponibles para cualquier controlador de dispositivo de modo real que esté cargado.
- **1 − 4 MB:** normalmente no se utilizan, Windows NT se carga por encima de esta dirección y, para mantener la compatibilidad. Windows 95 no utiliza estos espacios de direcciones, como tampoco lo hacen las aplicaciones de 32 bits basadas en Windows. No obstante, estos espacios de direcciones están disponibles para que lo utilicen las aplicaciones de 16 bits basadas en Windows.
- **4 MB − 2 GB:** las utilizan las ampliaciones de 32 bits basadas en Windows (y algunas de 16 bits).
- **2 − 3 GB:** las utilizan las DLL y otros objetos compartidos. Por ejemplo: los cuadros de diálogo estándar que utilizan todas las aplicaciones se encuentran en las DLL de diálogos comunes. COMMDLG.DLL y COMMDLG32.DLL. Estos archivos se cargan en este espacio de direcciones.
- **3 − 4 GB:** están reservadas para uso del sistema operativo (todos los componentes del anillo 0 se asignan aquí). Por ejemplo: los controladores de video y otros controladores virtuales se cargan en este espacio de direcciones.

Paginación Excesiva

Cuando una dirección virtual señala una página que no se encuentra en la memoria física, el procesador genera un fallo de página. Dicho fallo indica al administrador de memoria virtual que cargue en la memoria la página que se encuentra en el disco. Si no hay suficiente memoria disponible para cargar la página, el administrador de memoria virtual debe sacar una página de la memoria para almacenarla en el disco.

El sistema está diseñado para permitir una cierta capacidad de paginación. Sin embargo, si se pagina frecuentemente, puede llegar un momento en el que el sistema pagine tanto que apenas quede tiempo para hacer trabajo útil. La paginación excesiva degrada el rendimiento global del sistema y puede ocasionar una avería prematura del disco duro.

Por lo general la solución a la paginación excesiva consiste en agregar más RAM al equipo o en trabajar con menos aplicaciones simultáneamente.

PLUG AND PLAY

Definición: es tanto una filosofía de diseño como un conjunto de especificaciones sobre la arquitectura de PC's. Se trata de un conjunto independiente de especificaciones desarrollado por un grupo de fabricantes de hardware y de empresas de software.

Cualquier empresa dedicada a la producción de equipos puede seguir la norma Plug & Play.

La finalidad de Plug & Play es permitir el cambio de la configuración de un equipo sin que sea necesaria la intervención del usuario. La instalación de cualquier dispositivo debe ser una tarea sencilla y libre de errores.

Para los dispositivos que se ajustan a esta arquitectura, la instalación es automática: se enchufa el dispositivo, se enciende el sistema y a trabajar. Con un sistema totalmente compatible con la arquitectura Plug & Play, el usuario deber ser capaz de cambiar los componentes del sistema sin tener que reiniciar el equipo. Pro ejemplo, un usuario podría hacer lo siguiente sin reiniciar la estación de trabajo ni cambiar parámetros de configuración:

- Insertar y quitar dispositivos (tarjetas de PC).
- Conectarse y desconectarse de una estación de acoplamiento.
- Conectarse y desconectarse de una red.

El sistema determina cuál es la configuración óptima para cualquiera de los cambios y las aplicaciones se ajustan automáticamente con el fin de aprovechar al máximo la nueva configuración. Esto son algunos ejemplos de las posibilidades que ofrece un sistema totalmente compatible con la arquitectura Plug & Play:

- Si tiene un notebook con características de comunicación por infrarrojos y lo lleva a una sala en la que hay una impresora que también utiliza comunicación por infrarrojos, sus aplicaciones estarán preparadas para imprimir en esa impresora.
- Si agrega una tarjeta nueva a su equipo, podrá empezar a utilizarla en cuanto encienda el equipo.
- Para agregar un nuevo monitor, solo tiene que enchufarlo y encenderlo: automáticamente podrá beneficiarse de todas las posibilidades que le ofrezca.

Para que un sistema Plug & Play sea completamente automático, hay varias funciones que deben llevarse a cabo.

Autoidentificación y Autoespecificación de los dispositivos

Los dispositivos Plug & Play deben ser capaces de identificarse a sí mismos y de especificar sus posibilidades y sus requisitos de recursos. Esta información permite al sistema operativo determinar y establecer una configuración que funcione para todos los dispositivos del sistema, así como cargar los controladores de dispositivos apropiados sin que sea necesaria la intervención del usuario.

Instalar un dispositivo nuevo se convierte en algo tan sencillo como enchufarlo, encender el sistema y si es necesario, insertar un diskette cuando lo solicite el sistema. Si el controlador ya está en el sistema, el último paso no es necesario.

Cambios dinámicos en la configuración

Un sistema Plug & Play permite insertar dispositivos en cualquier momento. No es necesario que el usuario apague el sistema o lo reinicie. Esto significa que, cuando se inserta un dispositivo, el sistema operativo recibe una notificación acerca del nuevo dispositivo, de sus posibilidades y de sus requisitos. Entonces procede a cargar los controladores de dispositivos necesarios y establece una configuración que funcione. No se le pide al usuario que intervenga en el proceso de configuración a menos que los recursos que necesita el dispositivo no estén disponibles.

Se notifica a las aplicaciones acerca de estos eventos dinámicos para que puedan aprovechar las nuevas funciones disponible o bien para que no intenten utilizar dispositivos que ya no estén disponibles.

Compatibilidad con sistemas y periféricos existentes

La arquitectura Plug & Play es compatible con los sistemas y periféricos existentes (heredados) que estén instalados. Para conseguir esa compatibilidad, los componentes de la arquitectura Plug & Play se acomodan a la falta de mecanismos de informe de los dispositivos que no son Plug & Play. La información sobre dichos dispositivos se almacena en el sistema y los dispositivos que no puedan configurarse mediante el software reciben la mayor prioridad en la asignación de recursos.

Cuando se producen conflictes irresolubles entre los dispositivos, el sistema guía al usuario a través de las diversas opciones de configuración de los dispositivos.

La arquitectur Plug & Play es extensible a los nuevos tipos de dispositivos que puedan surgir en el futuro.

Independencia de los sistemas operativos y del hardware

La arquitectura Plug & Play es abierta e independiente de sistemas operativos y hardware específicos. Los componentes de esta arquitectura están basados en interfaces publicadas abstraídas a un nivel que permite a la arquitectura dar cabida a las diferentes arquitecturas de bus y de dispositivos que existen hoy en día, así como las que se diseñan en el futuro.

Tipos de acoplamiento

Windows 95 permite tres tipos de acoplamiento: previo apagado, en marcha reducida y en marcha.

- En el acoplamiento previo apagado es necesario apagar el equipo (o reiniciarlo) para proceder al acoplamiento o al desacoplamiento. La mayoría de Laptops funcionan de esta manera.
- En el acoplamiento de marcha reducida, puede llevarse a cabo el acoplamiento o el desacoplamiento cuando el equipo se encuentra en un modo de funcionamiento reducido (por ejemplo, en modo suspendido o de letargo).
- En el acoplamiento en marcha, puede hacerse el acoplamiento o el desacoplamiento con el equipo funcionando a plena potencia.

El acoplamiento en marcha tiene dos variedades distintas:

- **Expulsión automática:** una interfaz de software activa un mecanismo de expulsión motorizada. Esto permite a Windows 95 pedir al usuario que actúe si hay algún recurso abierto o que se está utilizando. El usuario puede, por ejemplo, guardar archivos antes de que el sistema lleve a cabo la expulsión.

Este acoplamiento recibe a veces el nombre de tipo VCR

- **Manual:** el usuario lleva a cabo el desacoplamiento sin que haya ningún tipode bloqueos de hardware. Puesto que el sistem no puede controlar cuando se hace el acoplamiento o desacoplamiento, es el usuario quien debe cerrar archivos o actuar de la forma pertienente antes de llevar a cabo el desacoplamiento para evitrar la pérdida de datos. Este tipo de acoplamiento se conoce también como modo sorpresa.

Los dispositivos que cambian de configuración basándose en el estado de acoplamiento incluyen casi todo: monitores, acceso a redes, unidades de disco duro, retirables: cualquier recurso que no sea necesario para reiniciar el sistema puede cambiar basándose en la configuración.

Para acoplar o desacoplar algunos dispositivos es necesario seguir unos pasos previos.

Si durante el funcionamiento hay un cambio en la configuración del sistema el componente en cuestión debe ser capaz de notificar al respecto al sistema operativo para que este pueda configurar el nuevo dispositivo. Las aplicaciones deben ser capaces de responder a los cambios de configuración para aprovechar las ventajas de los nuevos dispositivos, así como para dejar de llamar a dispositivos que se han quitado.

Cuatro recursos vitales

Para entender parte de lo que hace la arquitecrua Plug & Play es preciso comprender los conceptos básicos de arquitectura de hardware de los equipos.

Los periféricos se comunican con la CPU y entre ellos mismo de cuatro formas dferentes.

Solicitud de intrrupción (IRQ)

Cuando un periférico quiere notificar a un programa de algo que ha sucedido y que debe ser tratado por el software, suele hacerlo emitiendo una solicitud de interrupción. Estas son señales del hardware incorporadas en el equipo de las que se encarga un dispositivo llamado controlador programable de interrupciones (PIC).

Puesto que solo hay 16 solicitudes de interrupción con las que trabajar y puesto que la mayoría de periféricos utilizan una IRQ, durante los años se han desarrollado ciertas convenciones. A continuación se indica la utilización más común de la solicitud de interrupción

LÍNEA DE IRQ	**USO**
0	Cronómetro del sistema
1	Teclado
2	Cascada al segundo PIC
3	Puerto de comunicación 1
4	Puerto de comunicación 2
5	Puerto paralelo 2
6	Unidad de diskettes
7	Puerto paralelo 1
8	Reloj de tiempo real
9	Libre
10	Libre
11	Libre
12	Libre
13	Coprocesador matemático
14	Libre
15	Libre

Acceso Directo a memoria (DMA)

Para acelerar el acceso de los periféricos a la memoria, los equipos tienen 8 canales de acceso directo a memoria que permiten a los periféricos trabajar directamente con la memoria del sistema sin tener que hacerlo a través del procesador.

Puerto de entrada y salida (E/S)

Cuando un dispositivo quiere llevar a cabo una función de entrada o de salida, es normal que utilice una sección de la memoria llamada puerto de E/S.

Memoria

Muchos periféricos utilizan memoria interna o reservan memoria del sistema para ellos mismos. El mapa de memoria del equipo IBM PC original asignaba los 384 Kb superiores del espacio de direcciones de 1 MB para este tipo de memoria de dispositivos. Aunque algunos periféricos utilizan ahora la memoria de modo protegido que está por encima de la línea de 1 MB, la mayoría sigue asignando su mapa de memoria a esa área de 384 Kb que va del 000A0000 al 000FFFFF hexadecimal.

El modelo de árbol

Los equipos constan de diversas arquitecturas de bus y de dispositivos que coexisten en un sistema determinado. Puede ver estos sistemas utilizando un modelo de árbol que describe las relaciones entre los buses y los dispositivos.

En el árbol del sistema, los buses y los dispositivos pueden aparecer en diferentes niveles de la jerarquía del sistema, en diversas relaciones primario − secundario.

Los buses y los dispositivos pueden necesitar recursos del sistema y, a su vez, pueden facilitar recursos que pueden ser comprendidos o no por sus primarios.

Las relaciones se complican por el hecho de que algunos buses (SCSI, PCMCIA) se encuentran ocultos de sus primarios hasta que se inicialice.

Cada rama del árbol define un objetivo al que la estructura Plug & Play y qe se llama nodo de dispositivo. Para configurar un nodo de dispositivo se necesita de la siguiente información:

- **Un código de identificación exclusivo:** una cadena que describe el objetivo y que se utiliza durante diferentes faces del proceso Plug & Play.
- **Los requisitos de recursos del nodo de dispositivo:** cada uno de los requisitos debe indicar el tipo de recurso y las restricciones asociadas a este recurso específico.
- **Los recursos asignados al nodo de dispositivo:** ciertos dispositivos pueden necesitar IRQ específicas. Estas restricciones pueden tener interdependencias: por ejemplo, los puertos COM se configuran por lo general para utilizar la IRQ 3 y el puerto de entrada y salida 02f8 ò la IRQ 4 y el puerto de E/S 03f8.
- **Saber si el nodo de dispositivo es un bus:** si el dispositivo es un bus, el sistema debe identificar los nodos de dispositivos adicionales asociados a dicho bus y hacer un seguimiento de los recursos de los nodos de dispositivos.

El sistema operativo debe seguir con el proceso de configuración identificando cada nodo dispositivo del sistema y sus requisitos de recursos. Los dispositivos que no se inicializan (como un mouse) deben estar inactivos durante el encendido para que el sistema operativo pueda identificar cualquier posible conflicto entre los requisitos de dispositivos de los diferentes dispositivos antes de configurarlos.

El sistema operativo debe almacenar en un registro la información acerca de cada nodo de dispositivo y, luego, cargar los controladores de dispositivos correspondientes a cada nodo.

Proceso Plug & Play

Windows 95 implementa la identificación del sistema PnP de todos los dispositivos nuevos tanto durante el

inicio como de forma dinámica mientras el sistema está funcionando.

Cada vez que Windows 95 se inicia, se pide al hardware la información de identificación y de especificación.

Cuando se agrega al sistema un dispositivo PnP, dicho dispositivo pasa dinámicamente su información de identificación y especificación al sistema operativo Windows 95, y los componentes PnP de este, se encargan de todo lo relativo a la configuración.

Proceso PnP para los adaptadores ISA

La especificación correspondiente a los adaptadores ISA PnP definen un mecanismo de hardware y de software que se incorpora en la próxima generación de tarjetas ISA, a los que se conoce como tarjetas ISA PnP.

A continuación se indican los pasos principales del proceso de configuración automática:

- Poner todas las tarjetas ISA PnP en modo de configuración.

El software PnP identifica y configura los dispositivos mediante un conjunto de comandos que se ejecutan usando tres puertos de entrada y salida de 8 bits.

Se escribe una secuencia de datos en uno de los puertos (la clave de iniciación) que se utiliza para activar la lógica PnP de la tarjeta.

- Aislar una tarjeta ISA PnP cada vez y asignar un controlador a cada tarjeta.

Puesto que todas las tarjetas responden a las mismas direcciones de puertos de entrada y salida, el software PnP necesita un mecanismo de aislamiento para gestionar una tarjeta completa cada vez. El protocolo de aislamiento utiliza un número exclusivo de cada tarjeta (el ID del dispositivo y un número de serie) con el fin de aislar una tarjeta PnP cada vez.

Después del aislamiento, el software PnP asigna un controlador a cada tarjeta, el cual se utiliza para seleccionar esa tarjeta PnP concreta. El controlador elimina la necesidad de utilizar el protocolo de aislamiento, que es más elaborado y lleva más tiempo, para seleccionar una tarjeta específica.

- Leer la estructura de datos sobre los recursos de la tarjeta para determinar qué requisitos y posibilidades de recursos tiene la tarjeta.

Cada tarjeta ISA PnP posee una estructura de datos sobre los recursos, de sólo lectura, que describe los recursos a los que da soporte y que necesitan las funciones incluidas en ella.

- Asignar recursos que no sean conflictivos a cada tarjeta.

Se invoca un proceso de arbitraje de recursos para determinar qué recursos se asignan a cada tarjeta ISA. La configuración de la tarjeta ISA se lleva a cabo utilizando los registros de comandos especificados para cada tipo de recurso.

Después de haber asignado los recursos, puede invocarse un mecanismo de detección de conflictos de entrada y salida. Garantiza que los recursos de entrada y salda asignados no entren en conflicto con las tarjetas ISA estándar.

- Activar todas las tarjetas ISA PnP y retirarlas del modo de configuración.

Tarjetas heredadas

Es un sistema que utiliza únicamente tarjetas ISA PnP, es posible conseguir una configuración totalmente automática. Durante un período de transición, la generación actual (herencia) de tarjetas ISA estándar coexistirá con las tarjetas ISA PnP en el mismo sentido. En estos sistemas, la solución para la configuración ISA implica ciertas adiciones al BIOS y al sistema operativo para administrar y arbitrar los recursos de los buses ISA. En algunos casos seguirá siendo necesaria la intervención del usuario.

El sistema necesita saber el tipo de hardware existente y cómo está configurado.

Windows 95 lo determina utilizando un método de detección seguro o una rutina de sondeo invasora o bien pidiéndole al usuario que proporcione esta información.

Cuando se tiene toda la información, se graba en el registro. Esta información no es dinámica, pero se utiliza para arbitrar la asignación de recursos a los dispositivos que son compatibles con la arquitectura PnP.

Puesto que los dispositivos heredados tienden a tener menos flexibilidad en lo que se refiere a requisitos de recursos, se le asignan los recursos antes que a los dispositivos PnP

Componentes Plug & Play

Para implementar un sistema totalmente compatible con la arquitectura PnP es necesario realizar ciertos cambios en todos los componentes del equipo, entre los que se incluyen los siguientes:

- Enumeradores de bus.
- BIOS.
- Árbol de hardware y registro.
- Sistema operativo.
- Dispositivos y controladores de dispositivos.
- Árbitro de recursos.
- Administrador de configuración.

Enumeradores de bus: son los responsables de componer el árbol de hardware en un sistema Plug & Play. Son un nuevo tipo de controlador que se necesita para cada tipo específico de bus.

- El controlador está diseñado para comprender los detalles de implementación de una arquitectura de bus específica, lo que le permite identificar los dispositivos de dicho bus, leer sus requisitos de recursos y configurarlos siguiendo las instrucciones del administrador de configuración.
- Estos controladores pueden utilizar otros existentes o los servicios del BIOS para tener acceso al hardware. Por ejemplo:

TIPO DE DISPOSITIVO

ENUMERADOR

PCI

PCI.VXD

EISA

EISA.VXD

ISA Plug & Play

ISAPNP.VXD

Bus SCSI

Controlador de SCSI

PC Card (PCMCIA)

Controladores de servicios de tarjetas

Estos controladores también pueden implementarse en el BIOS para buses específicos como la tarjeta del sistema.

La función primordial del enumerador consiste en asignar un código de identificación exclusivo a cada uno de los dispositivos del bus. El único requisito que debe cumplir dicho código es que sea exclusivo y coherente. Cada vez que se inicia el sistema, el ID de un dispositivo concreto debe ser el mismo. La estructura PnP utiliza los códigos de identificación existentes para la mayoría de los buses.

Además, los enumeradores deben recuperar la información de configuración de los dispositivos ya sea directamente del dispositivo o del registro.

BIOS Plug & Play

Para cumplir los requisitos de la arquitectura PnP, se ha mejorado el BIOS del sistema de forma que proporcione los servicios de configuración de los dispositivos al iniciar y de notificación de los sucesos dinámicos. Dichas posibilidades deben estar muy integradas con el sistema operativo.

Configurar: como mínimo un BIOS PnP debe configurar los dispositivos de la tarjeta del sistema antes de pasar el control del proceso de configuración al sistema operativo. Este proceso implica el aislamiento y la inicialización de los dispositivos de la tarjeta del sistema (Controlador programable de interrupciones, controlador de DMA, controlador de sistema de video, controlador de diskettes, etc.)

Mantener los datos: en la estructura PnP, cada uno de estos dispositivos está asociado a un código de identificación exclusivo que el sistema operativo reconoce. El BIOS también mantiene un lista con información sobre la configuración de los dispositivos de la tarjeta del sistema y comunica dicha información al sistema operativo cada vez que se ha completado el proceso de autocomprobación de encendido (POST).

Notificar: para ofrecer las funciones PnP al completo, el BIOS notifica al sistema operativo los sucesos dinámicos de configuración como, por ejemplo, la inserción de un sistema notebook en una estación de acoplamiento. Un BIOS PnP proporciona el mecanismo para que el sistema operativo PnP reconfigure los dispositivos de la tarjeta del sistema como respuesta a un suceso dinámico. Esto permite al sistema operativo volver a configurar el sistema sin que el usuario tengo que apagarlo primero. El sistema operativo también puede notificar a las aplicaciones y a los controladores de la nueva configuración. Además, para los dispositivos controlados por software el BIOS puede avisar con prontitud al sistema operativo y evitar errores y pérdida de datos cuando se quita el dispositivo.

Si el BIOS no es compatible con la arquitectura PnP, Windows 95 utiliza pruebas invasivas para determinar el tipo de bus y el estado, o bien solicita la información al usuario.

El árbol de hardware y el registro

El árbol de hardware es un registro en RAM de la configuración actual del sistema. La información del árbol se obtiene del registro de información de configuración de todos los dispositivos, tanto si están instalados actualmente como si no. (en Windows 95 esta información se guarda en SYSTEM.DAT). EL árbol de hardware se crea cada vez que se inicia el sistema y se actualiza cuando tiene lugar un cambio en la configuración del sistema durante el tiempo de ejecución.

El formato de este árbol define un esquema estándar para identificar cada dispositivo, así como sus requisitos y restricciones en cuanto a recursos, si los tiene. También pude hacer independencias en ciertos recursos. Las aplicaciones y los controladores pueden tener acceso al registro y obtener información acerca de configuraciones alternativas, el software necesario para que funcionen los dispositivos y las configuraciones definidas por el usuario. La existencia de esta base de datos hace que los archivos *.INI específicos de muchos dispositivos y las aplicaciones y ano sean necesarios.

En Windows 95, esta información se almacena en el registro bajo HKEY_LOCAL_MACHINE.

Sistema operativo Plug & Play

Para ofrecer todas las funciones de la arquitectura PnP, el sistema operativo necesita ciertos componentes nuevos, que son los siguientes:

- **El administrador de configuración:** software que controla el proceso de configuración y mediante el cual pude establecerse la comunicación con todos los componentes que participan en dicho proceso.
- **Al árbol de hardware:** base de datos con información, almacenada en la memoria RAM, que se utiliza para configurar el sistema.
- **Enumeradores de bus:** controladores para identificar todos los dispositivos de un bus concreto así como sus requisitos de recursos.
- **Árbitros de recursos:** controladores que asignan recursos a todos los dispositivos.

Además, deben modificarse varios componentes, incluyendo el programa de instalación del sistema operativo y la interfaz de usuario.

Controladores de dispositivos

Basándose en los modelos de controladores de dispositivos existentes, la arquitectura PnP proporciona las API adicionales necesarias para la configuración de los dispositivos PnP. Para que sea posible reconfigurar el sistema y utilizar la memoria de forma más eficaz, es necesario que los controladores de dispositivos puedan cargarse y descargarse dinámicamente. Los controladores deben comunicarse con otros componentes del sistema de diversas formas, y deben ser capaces de:

- Registrarse ante el administrador de configuración cuando se cargan por primera vez.
- Permanecer inactivos hasta que se les asignen los recursos pertinentes.
- Comunicarse con las aplicaciones para responder a los sucesos dinámicos de configuración.

Árbitro de recursos

Es la parte del sistema operativo que asigna tipos específicos de recursos a los dispositivos y que resuelve los conflictos entre los dispositivos que solicitan los mismos recursos.

Si diferentes dispositivos necesitan los mismos recursos, los dispositivos deben ser capaces de proporcionar al sistema operativo información acerca de los requisitos de recursos alternativos. El sistema operativo utilizará

esta información para identificar una configuración que funcione. Una vez que se haya resuelto el conflicto de recursos, el sistema operativo debe almacenar la nueva información de configuración en el registro y notificar a los controladores de dispositivos de la nueva asignación de recursos.

Para realizar esta función, el árbitro de recursos contiene toda la información sobre cómo está estructurado un recurso, así como los algoritmos para determinar una configuración de recursos viable para un conjunto determinado de requisitos y restricciones.

El árbitro de recursos interactúa estrechamente con el administrador de configuración para llevar a cabo el proceso iterativo de asignación de recursos:

- Durante el encendido
- Como respuesta a sucesos dinámicos de configuración

En el caso de una reconfiguración dinámica del sistema, el administrador de configuración puede pedir al árbitro de recursos que libere recursos y los vuelva a asignar para satisfacer las necesidades de un dispositivo nuevo.

La separación funcional del árbitro de recursos y el administrador de configuración favorece la ampliación futura del sistema operativo para dar cabida a nuevos tipos de recursos.

Estos son los cuatro recursos arbitrados:

- Líneas de solicitud de interrupción (IRQ).
- Canales de acceso directo a memoria (DMA).
- Direcciones de memoria.
- Direcciones de puertos de E/S.

Administrador de configuración

El administrador de configuración coordina todo el flujo de operaciones llevadas a cabo por todos los componentes implicados en la configuración.

Este toma el control del proceso de configuración en dos casos:

- Cuando recibe del BIOS la lista de configuración de los dispositivos de la tarjeta del sistema.
- Cuando el BIOS o el enumerador del bus envía la notificación de un suceso dinámico de configuración.

El administrador de configuración coordina entonces la comunicación entre los enumeradores de bus, el árbol de hardware, el registro, los controladores de dispositivos y los árbitros de recursos para establecer la configuración del sistema apropiada. Además, notifica a los controladores de dispositivos y a las aplicaciones cualquier cambio presente o pendiente en la distribución del sistema.

Para llevar a cabo esta tarea, el administrador de configuración pide a los enumeradores que identifiquen todos los dispositivos de sus buses y sus respectivos requisitos de recursos y, luego, almacena ésta información en el árbol de hardware como una disposición jerárquica de nodos de dispositivos.

Se carga un controlador para dada dispositivo y el administrador de configuración le indica que espere la asignación de los recursos específicos. El administrador de configuración solicita a los árbitros de recursos que asignen los recursos a cada dispositivo y, si hay algún conflicto, lleva a cabo un proceso interactivo de reconfiguración hasta que determina una configuración que funcione. Para completar el proceso de configuración se informa a los controladores de dispositivos de la configuración de los dispositivos. Cuando

el BIOS o uno de los otros enumeradores de bus informa al administrador de configuración acerca de un suceso que precisa un cambio en la configuración del sistema, como quitar o insertar una PC Card, el proceso se inicia de nuevo.

Otros cambios en los componentes del sistema operativo

Parte del registro se crea mientras está en marcha el programa de instalación del sistema. Durante la inicialización, el sistema operativo llama a varios módulos de detección y de enumeración para que lleven a cabo un inventario de todos los dispositivos que hay en el sistema y registren la información sobre dichos dispositivos en la base de datos de configuración.

Intervención manual: en circunstancias normales, el sistema realiza todas las operaciones de configuración. Sin embargo, en ciertos casos puede que sea precisa la intervención del usuario. Por ejemplo, si se está instalando un dispositivo heredado, y el sistema no consigue detectarlo, puede forzar una instalación por software utilizando el ícono agregar nuevo hardware del panel de control

Puede que a veces el sistema no sea capaz de generar una configuración satisfactoria para un dispositivo. En este caso se necesitan los componentes de la interfaz del usuario para explicar al usuario lo que está pasando y preguntarle qué quiere hacer para resolver el problema. En Windows 95, las pantallas de solución de problemas de hardware de la ayuda indican al usuario la forma de solucionar el problema paso a paso.

Editor de configuración: puesto que el sistema PnP ofrece una forma de cambiar la configuración del sistema, éste debe proporcionar una herramienta especializada para tener acceso a esa información y modificarla, lo cual está pensado principalmente para los usuarios avanzados.

En Windows 95, dicha herramienta es el administrador de dispositivos.

Dispositivos Plug & Play

Hay muchos dispositivos compatibles con las especificaciones PnP; sin embargo, el grado de cumplimiento de estas especificaciones varía según el tipo de dispositivo.

SCSI: es una interfaz encadenada que se utiliza en muchos dispositivos, como unidades de disco duro y de CD − ROM.

Se ha completado una especificación SCSI PnP que permite los cambios dinámicos además de otras funciones PnP.

Para ofrecer la misma funcionalidad PnP, los dispositivos SCSI precisan tanto cambios de diseño como mejoras generales que faciliten su utilización. La configuración de un sistema SCSI puede dividirse en dos procesos distintos:

- La configuración del propio bus SCSI, como la terminación de ambos extremos del bus SCSI y el establecimiento del ID de los dispositivos.
- La configuración del adaptador de host SCSI, como la asignación de un canal IRQ, un canal DMA, etc.

Configurar un bus SCSI resulta difícil par la mayoría de los usuarios. Con uno de estos buses completamente compatible con la arquitectura PnP, el usuario sólo tendría que enchufar el nuevo dispositivo y el bus se ajustaría automáticamente. Pero con un bus SCSI actual, la lista de lo que el usuario debe afrontar para configurar un bus SCSI es larga, e incluye lo siguiente:

- Asignación de los ID de los dispositivos SCSI.
- Terminación.
- Paridad SCSI.
- Conjuntos de comandos.
- Geometría del disco y software.

La especificación SCSI 2 no define ningún mecanismo automatizado de asignación de ID, por lo que es responsabilidad del usuario garantizar que no haya dos dispositivos SCSI en el mismo bus SCSI que compartan el mismo ID SCSI.

Para los usuarios que no son técnicos no es fácil decir si un dispositivo SCSI está o no terminado, ya que la terminación puede hacerse mediante puentes o conmutadores, o bien, quitando o agregando resistores.

Actualmente, los usuarios pueden sustituir un adaptador de host SCSI por el de otra empresa y descubrir que no funciona debido a diferencias en la geometría de los discos o en la forma en que los dispositivos están asignados a los parámetros INT13h. Para que los buses SCSI ofrezcan toda la funcionalidad de la arquitectura PnP, es necesario solucionar todos estos asuntos.

Los adaptadores de host SCSI ISA completamente compatibles con la arquitectura PnP deben ajustarse a la especificación ISA PnP, por la que el hardware debe fabricarse de forma que los recursos como las líneas de IRQ y los canales DMA puedan obtenerse de forma automática, de manera que los usuarios no necesitan preocuparse sobre cómo asignar recursos del sistema y cómo garantizar que la asignación esté coordinada con el resto del sistema.

VESA Local Bus y PCI

El bus VESA local (VL) es un bus estándar que permite una conexión a alta velocidad con los periféricos.

PCI se trata de un bus local que pretende ser el sucesor del bus VL. Este es el tipo de dispositivo que se utiliza en la mayoría de PC's con Pentium y en el Apple Power Macintosh.

La arquitectura actual del bus PCI cumple la mayoría de los requisitos para ofrecer todas las funciones PnP. Estos dispositivos utilizan un esquema de identificación estándar y un mecanismos para indicar cuáles son sus requisitos en cuanto a recursos. El BIOS de las plataformas PCI contiene la lógica de configuración que se necesita para configurar todos los dispositivos PCI durante la autocomprobación de encendido (POST), y para activa sólo los dispositivos de inicialización y los dispositivos con ROM de opciones. Además, el BIOS ya proporciona un mecanismo para que los controladores de dispositivos accedan a los datos de configuración PCI.

El enumerador de bus PCI crea el árbol de hardware de los dispositivos PCI y los puentes PCI en el sistema operativo para que configure correctamente los sistemas que contienen dispositivos PCI e ISA. Puesto que los dispositivos PCI pueden compartir interrupciones, pero los ISA no, el BIOS debe ser capaz de impedir que el sistema operativo asigne los dispositivos PCI a interrupciones utilizadas por dispositivos ISA, que no pueden compartir las interrupciones.

Los dispositivos de los buses VL no son tan compatibles con las especificaciones PnP como los buses PCI. Tienen integradas rutinas de identificación y configuración y, puesto que corresponden a un bus dependiente de ISA, pueden nivelar gran parte del trabajo realizado con la especificación ISA.

Controladores IDE

Los controladores IDE ya permiten varias unidades de disco, por lo que si se agrega un esquema de

identificación y un mecanismo para indicar cuáles son los requisitos en cuanto a recursos, el estándar IDE puede ofrecer una solución económica para agregar unidades de CD − ROM PnP.

Especificación ECP (puerto de funcionalidad ampliada)

La especificación ECP puede ampliarse para abarcar los dispositivos PnP que utilizan el puerto paralelo del equipo y, con el desarrollo de un estándar para los puertos serie, también podría utilizarse el puerto serie para agregar dispositivos PnP al sistema.

Nuevas tecnologías

En el caso de las tecnologías en evolución, como las comunicaciones por infrarrojos, la arquitectura PnP ofrece un medio de mejorar la facilidad de uso y el atractivo de la tecnología.

Cambio de la configuración de los dispositivos PnP

En ciertas situaciones puede ser necesario cambiar manualmente la configuración de un dispositivo. Aunque la información acerca del dispositivo está almacenada en el registro, no debe cambiar dicha información, ya que la probabilidad de cometer errores (que además llevan a otros errores) es demasiado alta. Hay un componente de la interfaz de usuario que permite cambiar la configuración de los dispositivos.

Administrador de dispositivos

Es la interfaz de usuario que se utiliza para cambiar manualmente la configuración de los dispositivos.

- Para tener acceso al Administrador de dispositivos

- Desde el panel de control, hacer doble clic en el ícono Sistema, o hacer clic con el botón derecho del mouse sobre el ícono de Mi PC y elegir propiedades.

Aparecerá la hoja de propiedades del sistema.

- Hacer clic en la pestaña Administrador de dispositivos.

Esto le permite cambiar la configuración de cualquier dispositivo de la estación de trabajo. Puede ver los dispositivos por tipo o bien por conexión.

Verlos por conexión es tal vez más útil, ya que muestra mayor similitud con la información del registro. Además, si no se sabe dónde buscar un dispositivo pero sí se sabe de qué tipo de dispositivo se trata, puede encontrarlo por la conexión.

WINDOWS 9x,

22