

Administración de archivos en Windows

Windows incluye soporte para los siguientes sistemas de archivos:

CDFS UDF FAT12, FAT16 y FAT32 NTFS

Tipos de sistemas de archivos soportados

Cada uno de estos formatos es el más adecuado para ciertos ambientes, como se verá en las siguientes secciones.

CDFS: CDFS (\ Windows \ System32 \ Drivers \ Cdfs.sys), o sistema de archivos de CD-ROM, es un driver de sistema de archivos de solo-lectura que soporta el superconjunto del formato ISO-9660, como así el superconjunto del formato de disco Joliet. Mientras que el formato ISO-9660 es relativamente simple y tiene limitaciones como los nombres ASCII en mayúsculas con una longitud máxima de 32 caracteres, Joliet es más flexible y compatible con nombres Unicode de longitud arbitraria, por ejemplo. Si las estructuras para ambos formatos están presentes en un disco (para ofrecer la máxima compatibilidad), CDFS utiliza el formato Joliet. CDFS tiene una serie de restricciones:

- Un tamaño máximo de archivo de 4 GB
- Un máximo de 65.535 directorios
- CDFS se considera un formato anterior heredado por razones de compatibilidad, porque la industria, actualmente ha adoptado el formato de disco universal (UDF) como el estándar para los medios de sólo lectura.

UDF: La implementación Windows UDF de sistema de archivos, es OSTA (Optical Storage Technology) compatible con UDF. UDF es un subconjunto del formato ISO-13346 con extensiones de formatos como el CD-R y DVD-R/RW. OSTA define UDF en 1995 como un formato para reemplazar el formato ISO-9660 para los medios de almacenamiento magneto-ópticos, principalmente DVDROM. UDF está incluido en la especificación de DVD y es más flexible que CDFS. El controlador UDF admite las versiones de UDF 1.02, la versión 1.5 en Windows 2000 y versiones 2.0 y 2.01 en Windows XP y Windows Server 2003. El sistema de archivos UDF tiene las siguientes características:

- Los nombres ASCII para archivos y directorios, pueden ser de hasta 254 o 127 para caracteres Unicode.
- Los archivos pueden tener dispersión.
- Los tamaños de archivo se especifican mediante un código binario de 64-bits.
- Aunque el formato UDF fue diseñado teniendo en mente un soporte regrabable, el controlador de Windows UDF (\ Windows \ System32 \ Drivers \ Udfs.sys) dispone soporte para sólo lectura. Además, Windows no implementa el uso de otras características UDF, incluyendo secuencias (streams) con nombre, listas de control de acceso, o atributos extendidos.

FAT12: Windows es compatible con el sistema de archivos FAT principalmente por compatibilidad con las actualizaciones desde otras versiones de Microsoft Windows, por compatibilidad con otros sistemas operativos en sistemas de arranque múltiple, y compatibilidad con disquetes. El driver del sistema de archivos FAT, es implementado \ Windows \ System32 \ Drivers \ Fastfat.sys.

El nombre de cada formato FAT se debe al número de bits que usa dicho formato para identificar cada cluster en el disco. 12-bits para el identificador de clúster en FAT12, limita una partición para almacenar un máximo de 212 (4096) clusters. Windows utiliza tamaños de clúster de 512 bytes a 8 KB de tamaño, lo que limita un tamaño de volumen FAT12 a 32 MB (4096 x 8 x 1024). Por lo tanto, Windows utiliza FAT12 como el formato para todos los discos floppy de 5 pulgadas y los disquetes de 3,5 pulgadas, que almacenan hasta 1,44 MB.

Nota: Todos los tipos de sistemas de archivos FAT, reservan los dos primeros clusters y los últimos 16 clusters de un volumen, por lo que el número de clusters que pueden emplearse para un volumen FAT12, por ejemplo, es ligeramente inferior a 4096.

FAT16: con un identificador de clúster de 16 bits, puede tratar 2¹⁶ (65536) clusters. En Windows, los tamaños de clúster de FAT16 van desde 512 bytes (el tamaño de un sector) a 64 KB, lo que limita el volumen FAT16 a tamaños de 4 GB. El tamaño de clúster que Windows utiliza depende del tamaño del volumen. Los distintos tamaños se muestran en la Tabla. Si se da formato a un volumen FAT que es inferior a 16 MB utilizando el comando format o el Administrador de Discos, Windows utiliza el formato FAT12 en vez de FAT16.

Tamaño del volumen Tamaño de cluster

Tamaño del volumen	Tamaño de cluster
0–32 MB	512 bytes
33 MB–64 MB	1 KB
65 MB–128 MB	2 KB
129 MB–256 MB	4 KB
257 MB–511 MB	8 KB
512 MB–1023 MB	16 KB
1024 MB–2047 MB	32 KB
2048 MB–4095 MB	64 KB

Un volumen FAT se divide en varias regiones, que se muestran en la Figura. La tabla de asignación de archivos, que da el nombre al sistema de archivos FAT, tiene una entrada para cada clúster del volumen. Debido a que la tabla de asignación de archivos es fundamental para la interpretación satisfactoria de los contenidos de un volumen, el formato FAT mantiene dos copias de la tabla, de manera que si un controlador del sistema de archivos o un programa de control de coherencia (como Chkdsk) no pueden acceder a una copia de la tabla FAT (a causa de un sector de disco malo, por ejemplo) se puede leer la otra copia.

La estructura y el uso de la tabla de FAT para FAT16, es como ya fue explicado al inicio de la unidad, cuando hicimos referencia a este formato en forma genérica.

FAT32: es el formato más recientemente definido para el sistema de archivos FAT, y es incluido en Windows 95 OSR2, Windows 98 y Windows Millennium Edition en sus inicios. FAT32 utiliza identificadores de 32-bit para cada clúster, pero se reservan los 4 bits más altos, así que solo se usan 28-bits por cluster. Debido a que los tamaños de los clústers en FAT32 pueden ser tan grandes como 32 KB, FAT32 tiene una capacidad teórica de 8-terabyte (TB) para un volumen. Sin embargo, Windows trabaja con volúmenes FAT32 compatibles con otros sistemas operativos, que limita los volúmenes FAT32 a un máximo de 32 GB. La Tabla siguiente muestra los tamaños predeterminados de clúster para volúmenes FAT32.

Tamaño de la partición	Tamaño del cluster
32 MB–8 GB	4 KB
8 GB–16 GB	8 KB
16 GB–32 GB	16 KB
32 GB	32 KB

Una ventaja de FAT32 sobre los otros sistemas FAT, es que no tiene limitado el tamaño del directorio raíz ya que es más flexible su localización y posible crecimiento. También almacena una segunda copia del sector de boot para proporcionar tolerancia de fallo. Sin embargo tiene la limitación del tamaño de los archivos hasta un máximo de 4 GB debido a que el número de bits que se utiliza para el tamaño, solo tiene 32 bits ($2^{32} = 4\text{GB}$).

Sistema de archivos NTFS

El sistema de archivos NTFS se introdujo con la primera versión de Windows NT, y es totalmente distinto de FAT. Provee estructura de datos del sistema, que proporcionan mayor velocidad de acceso, y fundamentalmente mucha más seguridad, compresión archivo por archivo y encriptación. Es el sistema de archivos por defecto para nuevas instalaciones de Windows, y si se actualiza una versión antigua de Windows (como Windows NT con FAT 16), se preguntará si uno quiere convertir el sistema de archivos a NTFS.

Se puede convertir FAT16 o FAT32 a NTFS en cualquier momento. El problema será volver atrás ya que habría que reformatear el disco y será algo mas complicado.

Las ventajas que tiene sobre FAT son sobre todo la forma de tratar las estructuras de datos para mejorar el rendimiento, la fiabilidad y el uso del espacio en disco. Las mejoras de seguridad incluyen descriptores de seguridad que soportan listas de control de acceso.

NTFS utiliza una tabla de archivos maestra (MFT) para localizar todos los archivos dentro de un volumen NTFS. Todos los datos administrativos críticos son duplicados para permitir recuperación en caso de errores, asegurándose de que la pérdida en un sector no significará la pérdida de toda la partición. También puede recuperar datos de un sector dañado y asegurar de que ese sector no sea usado de nuevo.

NTFS incorpora muchas mejoras sobre el sistema FAT como compatibilidad mejorada con metadatos, y el uso de estructura de datos avanzadas (árboles-B) para optimizar el rendimiento, estabilidad, y el aprovechamiento del espacio en disco, además de nuevas características adicionales, como la seguridad, las listas de control de acceso o el registro de transacciones (journaling).

Gran parte de los datos administrativos que tiene que ver con los archivos, se almacena en forma de metadatos. Esto permitió una fácil ampliación de características de Windows NT. Un ejemplo lo hallamos en la inclusión de campos de indexado, añadidos para posibilitar el funcionamiento de Active Directory.

Los nombres de archivo son almacenados en Unicode (UTF-16), y la estructura del sistema de archivos (árbol de directorios) en árboles-B, una estructura de datos compleja que acelera el acceso a los archivos y reduce la fragmentación, que era lo más criticado del sistema FAT.

Se emplea un registro de transacciones (journal) para garantizar la integridad del sistema de archivos (pero no la de cada archivo). Los sistemas que emplean NTFS han demostrado tener una estabilidad mejorada,

que resultaba un requisito ineludible considerando la naturaleza inestable de las versiones más antiguas de Windows NT.

Sin embargo, a pesar de lo descrito anteriormente, este sistema de archivos posee un funcionamiento prácticamente secreto, ya que Microsoft no ha liberado su código como hizo con FAT.

Gracias a la ingeniería inversa, aplicada sobre el sistema de archivos, se desarrolló controladores como el NTFS-3G que actualmente proveen a sistemas operativos GNU/Linux, Solaris, MacOS X o BSD, entre otros, de soporte completo de lectura y escritura en particiones NTFS.

Un archivo NTFS no es una sucesión lineal de bytes como los archivos FAT o ext2 de Linux. Cada archivo está representado por múltiples atributos conocidos como flujos de bytes. Todos los archivos tienen básicamente dos flujos cortos para el nombre e información estándar y un flujo largo para el contenido. Sin embargo pueden tener varios flujos cortos y varios flujos largos.

Cada flujo (atributo) se representa mediante una cabecera (descriptor del atributo) y el cuerpo (datos del atributo). Cada flujo es un recurso que se puede bloquear en forma independiente. También cada flujo cuando se abre, tiene un puntero similar al puntero del archivo.

Cada flujo largo (datos) podría ser una versión diferente del mismo archivo o contener otros objetos asociados al archivo. En otras palabras es como tener varios archivos dentro de un archivo mayor.

Estructura del sistema de archivos

Describiremos a continuación un resumen del área administrativa de un disco lógico NTFS. En Windows se llaman volúmenes a los discos lógicos. Por lo tanto podemos decir que cada volumen esta organizado como una sucesión lineal de bloques o clusters de tamaño fijo para un determinado volumen. El tamaño del bloque depende del tamaño del volumen. A mayor volumen, mayor tamaño del bloque. Los tamaños de bloque varían entre 512 bytes y 64 Kbytes. Un valor común es de 4 Kbytes.

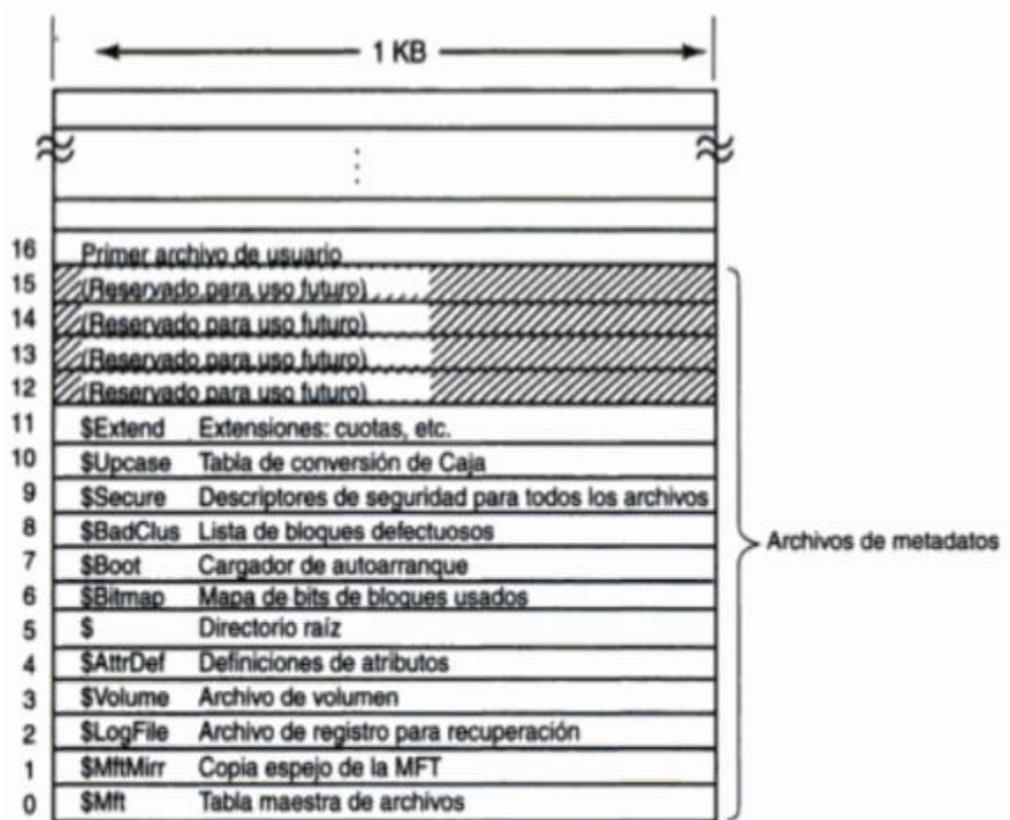
A continuación vemos la estructura de un volumen NTFS.

Sector de arranque	Tabla maestra de archivos	Archivos del sistema	Area de archivos
--------------------	---------------------------	----------------------	------------------

La principal estructura de datos administrativos es la tabla MFT (Master File Table) que permite registrar los atributos del archivo y punteros a los bloques, de los archivos del volumen. Cumple una función similar a la tabla de FAT, pero es mucho más completa y sofisticada.

La MFT es en realidad un archivo, y como tal está compuesto por atributos, donde el atributo específico que contiene los datos sería la tabla en si. Como todo archivo se puede expandir sobre bloques del disco ubicados en cualquier lugar físico de este y puede crecer hasta un tamaño máximo de 248 registros de 1KB.

En la siguiente figura se muestran los detalles de la tabla MFT.



Como se indica en la figura, cada entrada o registro de la tabla tiene 1 Kbyte y contiene básicamente atributos (metadatos) y punteros a los bloques de un archivo. Si los atributos tienen un contenido muy extenso o si el archivo tiene un gran contenido, se puede extender la descripción del archivo usando más registros mediante una lista enlazada de registros. El primer registro se llama Registro Base.

Todo archivo comienza en un registro base que se puede expandir mediante otros registros MFT, y si es extenso, se continúa con datos en bloques del disco fuera de la MFT, pero apuntados desde el o los atributos de datos de los registros MFT del archivo.

Los primeros 16 registros corresponden a archivos del sistema, conocidos como archivos de metadatos. Estos archivos tienen un formato y uso perfectamente definidos, aunque pueden cambiar en las futuras versiones. El nombre de estos archivos empieza con "\$" para indicar que se trata de un archivo de metadatos.

Los dos primeros registros describen a las dos copias de la MFT que en realidad son archivos que se pueden continuar en algún lugar del volumen y no necesariamente al comienzo de este. De esta forma se pueden evitar zonas del disco que estén dañadas. El primer registro describe al archivo MFT, de tal manera que su atributo de datos permite localizar a los bloques de datos de la MFT. El segundo es una copia del anterior que se almacena por seguridad. Como es necesario conocer físicamente cual es el primer bloque de la MFT, este se indica en el sector de arranque de la partición. Una vez localizado el primer bloque de la MFT, se puede navegar a través de esta mediante los enlaces provistos en la estructura contenida en tal bloque, y de esa forma localizar cualquier archivo del sistema de archivos.

☒ Los archivos \$Mft y \$MftMirr, se refieren a las copias de la tabla MFT como ya vimos. ☒ El archivo \$LogFile, se refiere al archivo de registro que contabiliza todos los cambios estructurales del sistema de archivos. Por ejemplo a la creación y eliminación de directorios y archivos, al cambio de atributos de los mismos, etc. Todos los cambios menos los cambios de contenido de los archivos. ☒ El archivo \$Volume, contiene información global del volumen como tamaño, rótulo,

versión, etc. El archivo AttrDef, define la estructura de los atributos que se pueden usar en cada archivo El archivo \$ es el directorio raíz, que como todo archivo puede crecer sin límite prácticamente El archivo Bitmap, se usa para registrar el uso o no de los bloques del disco El archivo \$Boot, se usa para localizar al programa de arranque El archivo BadClus, se usa para agrupar mediante una lista enlazada a todos lo bloques defectuosos, que deberán ser ignorados cuando se deba asignar un bloque a un archivo El archivo \$Secure permite definir todas las combinaciones de seguridad, y enlazar a cada una de ellas los archivos que posean tal seguridad El archivo \$Upcase, permite resolver la conversión entre mayúsculas y minúsculas en todos los alfabetos. El directorio \$Extend, permite acceder a archivos del sistema donde se registran cosas tales como cuota de disco, puntos de re-análisis, identificadores de objetos, etc.

Luego siguen cuatro registros que se mantienen para uso futuro.

Estructura de los registros



Cada registro comienza con una cabecera de registro, y luego le siguen pares de cabecera de atributos:valor. En la siguiente figura se muestra un ejemplo.

El encabezado de registro contiene un número mágico que se incrementa cada vez que se sobrescribe el registro (se re-utiliza para un nuevo archivo), un contador de referencia que indica cuantas estructuras administrativas están usando dicho registro, la cantidad de bytes usados en el registro y otros valores.

Después del encabezado le siguen pares de encabezado de atributo : datos administrativos y finalmente encabezado de atributo datos : datos. En la figura se muestra un archivo que solo necesita tres pares. En el se usan descriptores de los siguientes atributos: información estándar, nombre de archivo y datos. Sin embargo, NTFS define 13 tipos de atributos. Los datos son uno de estos atributos. A continuación indicamos estos atributos.

Información estándar: Bits indicadores, fechas y horas, etc. Son los atributos básicos. Nombre de archivo: En Unicote. Podría repetirse para nombre MS-DOS Descriptor de seguridad: Obsoleto. A partir de Windows 2000 esta información se registra en \$Secure Lista de atributos: Ubicación de registros MFT adicionales si hacen falta Identificador de objeto: Identificador de archivo. Es un número único en este volumen Punto de re-análisis: Para montajes y enlaces simbólicos Nombre del volumen: Nombre de este volumen Información del volumen: Versión del volumen. Solo se usa en \$Volume Raíz índice: se usa en directorios Asignación índice: Se usa en directorios muy grandes Mapa de bits: Se usa en directorios muy grandes Flujo utilitario de registro: Controla las entradas en \$LogFile Datos: Datos de flujos. Puede haber varios

Los datos consisten en una cabecera y los datos propiamente dichos, si estos caben en el registro. En caso contrario se indican pares de valores denominados series que consisten en: número del bloque de comienzo de una región contigua:longitud (cantidad de bytes).

Como vemos, NTFS trata de ubicar bloques contiguos para evitar el problema de la dispersión de bloques en el disco que en Windows se conoce como fragmentación del archivo.

En la figura anterior vemos la estructura típica de un archivo corto (que solo necesita un registro MFT). Consiste en un encabezado que describe una sucesión de bloques lógicos consecutivos. Si el archivo tiene huecos lógicos (hay ciertos bloques lógicos que no se usan) se usa una cabecera por cada sucesión de bloques lógicos. En esta cabecera se indica que la sucesión de bloques lógicos contiguos empieza con el bloque lógico 0 (o sea que la sucesión de bloques consiste en los bloques lógicos consecutivos 0,1,..,8, y que la próxima sucesión comenzaría con el bloque lógico 9. La serie #1 de bloques físicos comienza en bloque 20 y tiene 4 bloques (los bloques físicos 20, 21, 22 y 23 que corresponden a los bloques lógicos 0,1,2 y 3). La serie #2 comienza en el bloque físico 64 y contiene 2 bloques (los bloques físicos 64 y 65 que corresponden a los bloques lógicos 4 y 5). La serie #3 comienza en el bloque físico 80 y contiene 3 bloques (los bloques físicos 80, 81 y 82 que corresponden a los bloques lógicos 6, 7 y 8).

El número de series depende del tamaño del archivo y de lo bien que haya trabajado el asignador de bloques de disco para evitar la dispersión (bloques físicos no consecutivos).

En el ejemplo anterior se supuso un archivo corto que usa un solo registro MFT. ¿Que sucede cuando se necesitan más de un registro MFT para describir los bloques del archivo?

Simplemente en el primer registro (registro base) se incluye un atributo del tipo lista de atributos que permite apuntar a los otros registros MFT que se usarán como extensión. A su vez los registros MFT de extensión contendrán las sucesivas series de bloques del archivo. Si el registro base no puede contener al atributo lista de atributos, por ser demasiado extenso (el archivo es muy grande y hay muchos fragmentos), el atributo lista de atributos deja de ser residente (no contendrá la información de los punteros en si) y solo contendrá punteros a bloques del disco que si se usarán para contener los punteros).

Una entrada MFT para un directorio pequeño se muestra a continuación.



Registro MFT para un directorio pequeño.

Cada entrada de directorio (tiene un formato específico) contiene básicamente el nombre del archivo, la longitud del nombre y un puntero a la entrada MFT del archivo. La búsqueda de un archivo dentro del registro del directorio es secuencial en base al nombre del archivo. Una vez localizado, mediante el puntero se ubica el registro MFT del archivo y desde allí a los bloques del contenido del archivo.

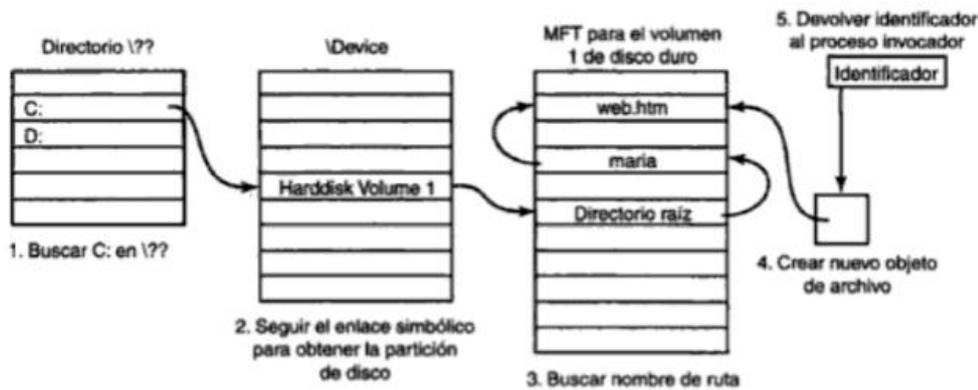
Los directorios grandes emplean un formato diferente. En lugar de enumerar los archivos de manera lineal, se utiliza una estructura de datos árbol B+ que permita realizar una búsqueda ordenada con máxima velocidad.

Navegación del árbol de directorios

Ahora explicaremos los pasos que se realizan para navegar la estructura del árbol de directorios para localizar un archivo partiendo de su nombre.

En la figura vemos la secuencia de operaciones del sistema operativo cuando recibe una llamada para abrir un archivo mediante la función:

```
d=CreateFile("c:\maria\web.htm", ...).
```



Pasos para buscar el archivo C:\maria\web.htm.

Esta llamada invoca una librería donde se antepone \?? Al nombre del archivo, quedando finalmente como \??\C:\maria\web.htm. Este nombre luego se pasa a la llamada del sistema operativo NtFileCreate.

Luego el sistema operativo usa al administrador de objetos para que comience la exploración. Primero se accede a la estructura raíz del espacio de nombre del administrador de objetos, que tiene el nombre \???. En dicha estructura localiza al archivo C:, que contiene un puntero a otra estructura del espacio de nombres del administrador de objetos en donde se localiza la información de la ubicación física del volumen, que a su vez tiene un puntero al directorio raíz de dicho volumen en la MFT correspondiente. Luego se localiza mediante el puntero correspondiente, en la entrada del directorio, al registro MFT del directorio maria. En este directorio se busca el nombre web.htm, que al ser localizado proporciona el puntero al registro MFT web.htm.

Localizado el archivo, el administrador de objetos crea un objeto sin nombre que contiene una propiedad con el puntero al registro base de web.htm. Finalmente devuelve en la variable d un identificador de dicho objeto que será compartido por todos los hilos del mismo proceso. Un hilo de dicho proceso usará dicho identificador cuando deba realizar la llamada ReadFile para leer dicho archivo. En este caso se accede al objeto y se obtiene la ubicación del registro MFT que contiene al archivo web.htm sin necesidad de explorar el árbol nuevamente.

Además de archivos regulares y directorios, NTFS también maneja archivos de tipo enlace como se usa en UNIX para enlaces blandos o simbólicos. Un enlace blando es simplemente un registro MFT que contiene un atributo punto de re-análisis que permite apuntar al atributo de datos del archivo. Dicho archivo podrá estar en el mismo u otro volumen.

Este mecanismo también puede usarse para el montaje de sistemas de archivos.

También es posible manejar enlaces duros como en UNIX ya que un archivo permite varios atributos de nombres.