HFS (MAC)

Hierachical File System

File Manager

Componente del sistema operativo que maneja el acceso a los archivos.

Finder

Parte del sistema operativo que maneja la búsqueda de archivos a ser presentados.

Archivos

Identificación de archivos

FileID: es un número identificatorio dentro del volumen asignado al archivo al ser creado. Es el número del nodo del catálogo del volumen de el archivo. Si se copia el archivo,este ID cambia pero si se mueve o renombra no

DirectoryID: Es similar al FileID. Se definen dos DirectoryID especiales:

- DirectoryID =1. Padre del directorio raiz. (en realidad no existe, es sólo por cuestiones de implementación)
- DirectoryID=2. Directorio raiz del volumen.

Como está formado un archivo

Todos los archivos están formados por dos partes llamadas FILE FORKS.

Resource fork: Contiene información sobre los recursos de un archivo. En el caso de una aplicación, contiene los menúes, iconos, cuadros de diálogo, código ejecutable. La información sobre estos recursos se guarda de forma estructurada.

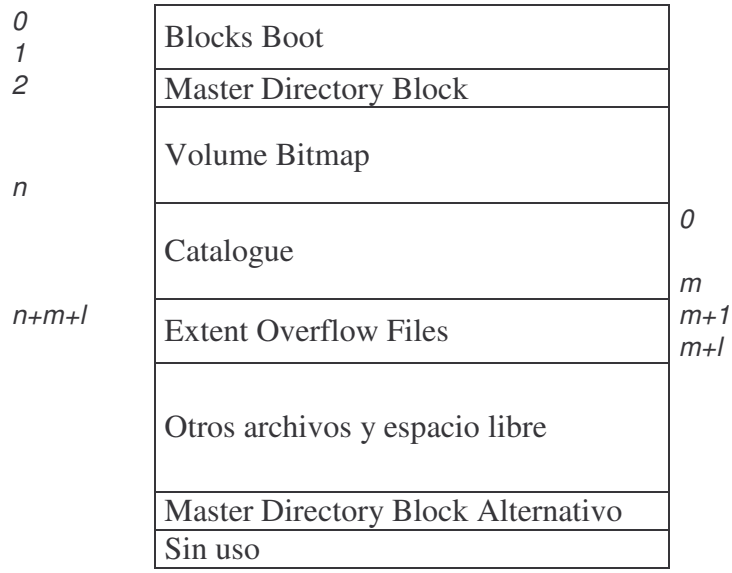
Data Fork: es una tira de bytes sin ningún formato predefinido. Es responsabilidad de la aplicación que haga uso del archivo interpretar estos datos. En el caso de un documento contendrá por ejemplo, el texto del mismo.
Si bien sería posible guardar datos en el resource fork, esto no es aconsejable por varias razones:

- La cantidad de recursos que puede contener es limitada
- La búsqueda en el resource fork es secuencial
- Sería necesario definirle a los datos una estructura que se ajuste a una definición de recursos

Un archivo de aplicación puede tener su data fork vacío y un docmento puede tener su resource fork vacío, pero siempre tiene ambas partes.

Estructura de un volumen HFS

Allocation Block: grupo de bloques contiguos que maneja es sistema operativo. Son la mínima unidad de disco direccionable. Su tamaño se establece al inicializar el file system. En volúmenes chicos, su tamaño puede coincidir con el del bloque lógico.
Clump: conjunto de bloques que se asigna cada vez que un archivo requiere mas espacio



Boot Blocks

Parámetros de configuración del sistema como la longitud de la cola de eventos, cantidad máxima de archivos abiertos.

Master Directory Block

Se escribe al inicializar el volumen

- Fecha y hora de creación del volumen
- Fecha y hora de última modificación del volumen
- Atributos del volumen (si está bloqueado por hardware, si fue desmontado correctamente.
- Cantidad de archivos en el directorio raíz
- Primer bloque del Volume Bitmap. Siempre vale 3 en esta implementación.
- Numero del allocation block en el cual comenzará la siguiente búsqueda para allocar.
- La cantidad de allocation blocks en el volumen
- El tamaño EN BYTES de un allocation block (múltiplo de 512)
- Tamaño de Clump por defecto
- Ubicación del primer allocation block del volumen
- Primer ID libre del catálogo
- Cantidad de allocation blocks sin usar en el volumen

- Nombre del volumen
- Fecha y hora del último backup
- Número de secuencia del backup
- La cantidad de veces que se escribió en el volumen
- Tamaño del clump para el Extents Overflow File
- Tamaño del clump para el Catalog File
- Cantidad de directorios en el directorio raiz
- Cantidad de archivos en el volumen
- Cantidad de directorios en el volumen
- Información usada por Finder
- El tamaño (en allocation blocks) del caché de volumen
- El tamaño (en allocation blocks) del caché del Volume Bitmap
- El tamaño (en allocation blocks) del extent Overflow File
- Primer Extent Record del Extent Overflow File. En los primeros 2 bytes guarda el número deAllocation block en el que se encuentra el primer nodo del Extent Overflow File
- Primer Extent Record del Catalog File. En los primeros 2 bytes guarda el número deAllocation block en el que se encuentra el primer nodo del Catalog File

Volume Bitmap

Contiene un bit por cada allocation block del volumen indicando si está asignado a algún archivo o no. La cantidad máxima es 65.535 (limitada por los 2 bytes sin signo del campo Cantidad de Allocation Blocks en el MDB) lo cual implica que el tamaño máximo de esta estructura es de 8192 bytes o sea 16 bloques lógicos.

Catalog File

Mantiene información sobre la jerarquía de archivos y directorios en un volumen. Está estructurado como un árbol B*. A cada nodo del catálogo se le asigna unCatalog Node ID (CNID) que es a su vez el FileID o DirectoryID del archivo o directorio.

Además de los ya vistos ID especiales para el raiz y su padre ficticio, los siguientes ID están predefinidos

3. FileID del Extents File
4. FileID del Catalog File
5. FileID del Bad Allocation Block File

Clave del Catalog File

La clave de búsqueda consiste en el nombre de archivo o directorio y el DirectoryID de su padre

Registros de Datos del Catalog File

Un nodo hoja del catálogo puede contener cuatro tipos de registros

- *Directory Records*: contienen información sobre un solo directorio
- *File Records*: Contiene información sobre un solo archivo
- *Directory Thread Records:* provee un link entre un directorio y su padre
- *File Thread Records:* provee un link entre un archivo y su directorio padre.

Extent Overflow Files

El File Manager mantiene una lista de segmentos contiguos de disco que pertenecen a cada. Cuando esta lista es muy larga, la información de algunos de esos segmentos (o extents) se almacena en el Extent Overflow File.
Este archivo también se organiza como árbol B*.

Extent Descriptor:

Primer Bloque
Cantidad de Bloques

Extent Record:

El File Manager lee los extents de a 3, usando un Extent Record, que no es mas que un array de 3 Extent Descriptors.

Master Directory Block alternativa

Sólo se actualiza cuando se agranda el Extent Overflow File o el Catálogo

Arboles B*

Los árboles B* usados por el File Manager sólo usan el data fork de los archivos en los que se encuentran almacenados. Se asigna un nodo por bloque de 512 Kb.

Estructura de un nodo

<table>
<tr><td rowspan="5">Node Descriptor</td><td>Puntero al sig. de este tipo</td></tr>
<tr><td>Puntero al ant.r de este tipo</td></tr>
<tr><td>Tipo de Nodo</td></tr>
<tr><td>Nivel del Nodo</td></tr>
<tr><td>Cant. registros</td></tr>
<tr><td colspan="2">Registro 0</td></tr>
<tr><td colspan="2">Registro 1</td></tr>
<tr><td colspan="2">Espacio libre</td></tr>
<tr><td colspan="2">Offset del espacio libre</td></tr>
</table>

Offset del registro 1
Offset del registro 0 (es siempre =)

Estructura de un registro de nodo (Node Record)

Contienen datos o un puntero a algún otro nodo del árbol.

Longitud de la clave (1 byte)	Clave de búsqueda (Máximo 255 bytes)	Datos o puntero

Tipos de nodos

Header Nodes

Es el primer nodo (0). Contiene siempre 3 registros.

Node Descriptor
B* Tree header record
Sin uso. Reservado
B* Tree map record
Offset del espacio sin uso
Offset al registro 2
Offset al registro 1
Offset al registro 0

B* Tree map record: Bitmap que indica que nodos del árbol están usados y cuales no.

B* Tree header record:

- Prof. actual del B*
- Nro. del nodo raiz
- Cant. de Hojas
- Nro. de la primera hoja
- Nro. de la última hoja
- Tamaño del nodo
- Longitud max. de clave
- Cant. de nodos en el árbol
- Cant. de nodos libres.

Map Nodes

Si el árbol B* tiene mas de 2048 nodos (alcanza para unos 8000 archivos), se usan Map Nodes para almacenar la parte del bitmap de nodos del árbol que no entra en el header record.

Index Nodes (Nodos Indice)

Contienen punteros en vez de datos, los cuales apuntan a otros nodos del árbol. Sirven para construir la jerarquía del árbol. Contienen Pointer Records (Clave + Puntero). El primer Index Node del árbol se llama Root Node o Nodo Raíz.

Leaf Nodes (Nodos Hoja)

Contienen datos. Su estructura depende del árbol B*. En el catálogo, será distinta de la del Extent Overflow File.