11.7 NTFS On-Disk Structure（8）

Indexing

In NTFS, a file directory is simply an index of file names—that is, a collection of file names (along with their file references) organized in a particular way for quick access. To create a directory, NTFS indexes the filename attributes of the files in the directory. The MFT record for the root directory of a volume is shown in Figure 11-43.

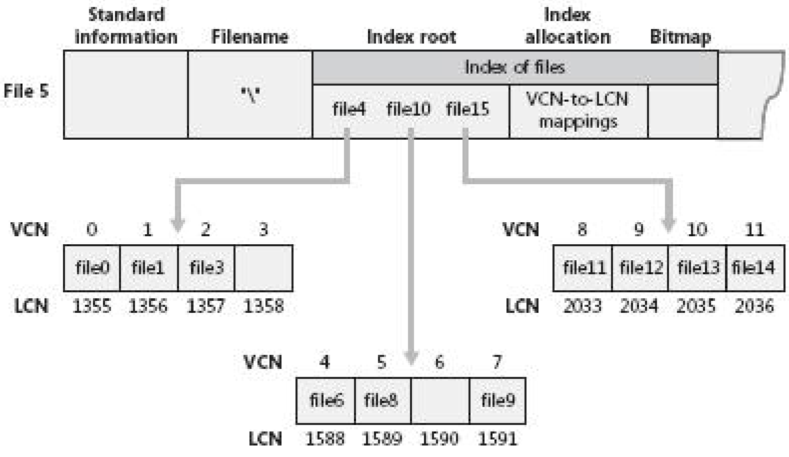

FIGURE 11-43 File name index for a volume's root directory

Conceptually, an MFT entry for a directory contains in its index root attribute a sorted list of the files in the directory. For large directories, however, the file names are actually stored in 4-KB, fixed-size index buffers that contain and organize the file names. Index buffers implement a b-tree data structure, which minimizes the number of disk accesses needed to find a particular file, especially for large directories. The index root attribute contains the first level of the b-tree (root subdirectories) and points to index buffers containing the next level (more subdirectories, perhaps, or files).

Figure 11-43 shows only file names in the index root attribute and the index buffers (file6, for example), but each entry in an index also contains the file reference in the MFT where the file is described and time stamp and file size information for the file. NTFS duplicates the time stamp and file size information from the file's MFT record. This technique, which is used by FAT and NTFS, requires updated information to be written in two places. Even so, it's a significant speed optimization for directory browsing because it enables the file system to display each file's time stamps and size without opening every file in the directory.

The index allocation attribute maps the VCNs of the index buffer runs to the LCNs that indicate where the index buffers reside on the disk, and the bitmap attribute keeps track of which VCNs in the index buffers are in use and which are free. Figure 11-43 shows one file entry per VCN (that is, per cluster), but file name entries are actually packed into each cluster. Each 4-KB index buffer can contain about 20 to 30 file name entries.

The b-tree data structure is a type of balanced tree that is ideal for organizing sorted data stored on a disk because it minimizes the number of disk accesses needed to find an entry. In the MFT, a directory's index root attribute contains several file names that act as indexes into the second level of the b-tree. Each file name in the index root attribute has an optional pointer associated with it that points to an index buffer. The index buffer it points to contains file names with lexicographic values less than its own. In Figure 11-43, for example, file4 is a first-level entry in the b-tree. It points to an index buffer containing file names that are (lexicographically) less than itself—the file names file0, file1, and file3. Note that the names file1, file3, and so on that are used in this example are not literal file names but names intended to show the relative placement of files that are lexicographically ordered according to the displayed sequence.

Storing the file names in b-trees provides several benefits. Directory lookups are fast because the file names are stored in a sorted order. And when higher-level software enumerates the files in a directory, NTFS returns already-sorted names. Finally, because b-trees tend to grow wide rather than deep, NTFS's fast lookup times don't degrade as directories grow. NTFS also provides general support for indexing data besides file names, and several NTFS features—including object IDs, quota tracking, and consolidated security—use indexing to manage internal data.