

# Windows OS File Systems

MS-DOS and Windows 95/98/NT/2000 allow use of FAT-16 or FAT-32.

Windows NT/2000 uses NTFS (NT File System)

## File Allocation Table (FAT)

### Disk Layout

See old Tanenbaum Fig 8-18.

Boot sector|Partition 1|Partion 2|...

Each partition can be a different file system (including Unix).

Each partition is laid out as:

Secondary Boot Sector|FAT|Optional Duplicate FAT|Root directory|Data blocks...

## File Allocation Table (FAT)

Have a table with one entry for each block on the disk.

Directory entry for a file contains the first block in the file.

The FAT entry for this block then points to the next block of the file.

Use EOF mark for last block.

Blank entries indicate free blocks (no need for free block list).

Example (also see old Tanenbaum Fig 8-19):

Block	Entry
0	Boot sector
1	Boot sector
2	3
3	7
4	Free
5	EOF
6	10
7	5
8	Free
9	Bad
10	EOF

The boot sector contains boot instructions and descriptive info like: size of disk sector, number of physical sectors on disk per block, size of root directory.

Bad entry indicates block is not usable.

With 16-bit blocks have a maximum of  $2^{16} = 64K$  blocks. To use a 2GB disk one would need 32K-byte blocks (very large). Can result in *internal fragmentation* for small files.

FAT-32 uses 32-bit block numbers supporting 4GB of block numbers and disks up to 2 Terabytes in size.

## FAT-16 Directory Entry

Each directory entry is 32 bytes:

- File Name (8 bytes)
- Extension (3 bytes)
- Attributes (1 byte)
- Reserved (10 bytes)
- Time (2 bytes)
- Date (2 bytes)
- Block number of first file block (2 bytes)
- File size (4 bytes)

Attributes (one-bit each):

R: Read Only

A: Archive (set when file modified, cleared when backed up)

S: System File (file cannot be deleted by the *del* command)

H: Hidden File (file is not listed with *dir* command)

D: Directory

V: Volume Label

Note: FAT and directory entry both keep track of how many blocks in a file—could be inconsistency.

FAT (versus Unix) keeps all information about a file in the directory entry rather than an inode.

## **FAT-32 File System**

Block numbers are 32-bits long. FAT itself becomes larger and two of the reserved bytes in directory entry are used to extend block number to 4 bytes.

FAT-32 also abolished the restriction on the size of the root directory—the root directory is no longer stored in a fixed position.

Like FAT-16, has no access protection for files.

## **Long File Names**

VFAT (virtual file allocation table) is used in Windows 95/98 to handle long file names. Use a reserved area of VFAT for long file names with an invalid combination of attribute bits.

# NT/2000 File System (NTFS)

## File System API Calls

Look at Tanenbaum Figures 11-31 and 11-33.

## Organization

Fundamental entity in NTFS is a volume. Like a partition in FAT, but occupy part, all or multiple disks.

Uses clusters (basically same as a block) as the unit of disk allocation. A cluster consists of one or more physical sectors on the disk. They are numbered by numbers called logical cluster numbers (LCNs).

Information about NTFS files stored as attributes: file name (names if there are aliases), creation time, security descriptor, unnamed data attribute (contents of the file).

## Master File Table (MFT)

Special file containing one entry for each file in a volume. MFT entry can be from 1KB to 4KB.

Info contained in an MFT entry about a file:

- standard info such as time stamps, file size
- file's name in Unicode (16-bit chars) and also an 8.3 DOS-style name
- security info
- data for the file. If a small file this attribute can be stored in the MFT entry (resident attribute vs. non-resident attribute). More details if file is larger in keeping track of which LCNs are used.

See Fig 11-35 for all attributes that can appear in MFT records.

## **MFT Directory Entry**

An MFT Directory entry is stored as a file and contains an index to the files in this directory. The index is stored as a B+ tree for fast lookup of file names.

Each file entry stores the file name, the number of its MFT entry, its time stamp and file size.

Latter two values are duplicated from file MFT entry for faster directory listings.

## **NTFS Volume Metadata**

NTFS volume metadata is stored in files:

- MFT itself
- copy of the MFT
- Log file records all metadata transactions to allow recovery from crashes
- Root directory
- Bitmap to show which clusters are allocated
- Boot file (to boot the system)
- Bad cluster file (clusters to avoid)

(Tanenbaum Fig. 11-34)

## **Other**

NTFS supports transparent file compression.

Also supports file encryption with public-key encryption