The early history of Windows file attributes, and why there is a gap between System and Directory

devblogs.microsoft.com/oldnewthing/20180830-00

August 30, 2018

Raymond Chen

Let's look at the values for the basic Windows file attributes. There's a gap where 8 should be.

`FILE_ATTRIBUTE_...`	Value
`READONLY`	`0x00000001`
`HIDDEN`	`0x00000002`
`SYSTEM`	`0x00000004`
	`0x00000008`
`DIRECTORY`	`0x00000010`
`ARCHIVE`	`0x00000020`

Rewind to CP/M.

CP/M supported eleven attributes:

Name	Meaning
F1, F2, F3, F4	User-defined
F5, F6, F7, F8	Interface-defined
T1	Read-only
T2	System
T3	Archive

The operating system imposed no semantics for user-defined attributes. You can use them for whatever you want.

The meanings of the interface-defined attributes were defined by each operating system interface. Think of them as four bonus flag parameters for each syscall that takes a file control block. You could set interface-defined attributes before calling a function, and that passed four additional flags in. Or the function could manipulate those attributes before returning, allowing it to return four flags out. Interface-defined attributes are always clear on disk.

The read-only bit marked a file as read-only.

The system bit had two effects: First, it hid the file from directory listings. Second, if the file belonged to user 0,[1] then the file was available to all users. (This was handy for program files.)

The archive bit reported whether the file has been backed up.

These attributes were retrofitted onto the existing directory structure by taking over the high bits of the eleven filename characters! That's why they are named F1 through F8 (high bits of the eight-character file base name) and T1 through T3 (high bits of the three-character extension, also known as the file type).

You can infer from this that CP/M file names were limited to 7-bit ASCII.

Anyway, MS-DOS 1.0 split the dual meaning of the system attribute into two attribute (hidden and system), and even though it didn't implement the read-only attribute, it reserved space for it.

That explains why the first three attributes are read-only (1), hidden (2), and system (4).

MS-DOS 2.0 most notably added support for subdirectories, but another feature that came along was volume labels. Since there was no space for the volume label in the disk header, the volume label was added as a directory entry in the root directory, with a special attribute that says "This is a volume label, not a file."[2]

The next attributes became volume label (8), directory (16), and archive (32).

Win32 adopted the same file attribute values as MS-DOS and 16-bit Windows, presumably in an effort to minimize surprises when porting code from 16-bit to 32-bit. The volume label attribute disappeared from Win32, but the bits for directory and archive were left at their original values to avoid problems with programs that operated with file attributes. Those programs contained their own definitions for the file attributes because 16-bit Windows didn't provide any.

[1] CP/M supported up to 16 users, numbered 0 through 15. When you started the computer, you were user 0, but you could change users by saying `USER n`. Files belonging to other users were inaccessible, except that system files belong to user 0 were available to everyone.

Anybody could change to any user at any time, so this was a file organization feature, not a security feature. In practice, nobody really used it because floppy discs were so small that it was easier to organize your files by putting them on different floppies than by trying to remember which user you used for each file.

[2] Windows 95 later repurposed the volume label attribute to mark directory entries as being used for long file names. Disk utilities often parsed directory entries directly, so any change in the disk format was a compatibility risk. The choice to use the volume label attribute for this purpose came after a lot of experimentation to find the least disruptive file format for long file names. It turns out that most low-level disk utility programs ignored anything marked with the volume label attribute.

Raymond Chen

Follow