The Write-Once ALTERNATIVE

by Simson L. Garfinkel

Is it stuck in the middle between CD-ROM and erasable, or is it unique in its value?

CD-ROM's READ-ONLY, INDELIBLE information storage capacity is already becoming established as a publication medium, and multiple-write or erasable magneto-optical systems seem to be just over the horizon. But what of WORM (write-once, read-many)? How does this so-called "middle" technology fit in, and who should use it?

Write-once systems offer several alternatives to other disc formats, including the potential as a publication format. Philips says its WORM discs will be able to be prerecorded at a factory. While other write-once formats, including CD-ROM, carry the same instrinsic ability, WORM differs in that it can be updated by the user—on-site or from a remote location—using a telephone and modem.

With WORM, sending a megabyte of information per month (about 200 pages) by modem can be accomplished in about an hour, and would meet most people's requirements for a financial information database or an entire index of career-related articles.

WORM can also be useful for making notes in the "margins" of a write-once "book." Just as a student underlines in a textbook or an engineer in a manual, a write-once document reading station could allow sections of a published work to be highlighted, changed, or noted upon. All changes would be stored on the disc.

Since all information written on a WORM, as with CD-ROM, is indelible, changes would not destroy existing information, nor could they be altered later.

Some would consider this feature a hindrance. An erasable disc, they argue, would allow users to delete obsolete data, yet important information could be protected by using software to indicate read-only files. While the argument is valid, it's also important to weigh WORM's large storage capacity: between 400MB and 1.2 gigabytes at present, enough to allow for reams of old material while leaving room for the new.

Many personal computer users adhere to the "packrat" approach to data storage where enormous libraries of 5.25-inch floppy disks are used as write-once storage media. This method stores only 360K per disk at a cost of 28 cents per kilobyte, or slightly less per kilobyte with high-density 1.2MB floppies. Meanwhile, a 400MB write-once disc costing $200 provides storage at about 50 cents per megabyte, or five cents per kilobyte.

Write-once systems also provide audit trails that cannot be forged—an important feature for business applications and medical records. If the manufacturer stamps a serial number on every disc, users can know with absolute certainty whether or not the retrieved data is original. For example, if John Mason's 1984 bank statement is stored on block #4221 of disc #8804330005, there is no way his record can be forged.

A final consideration in WORM's favor is its physical nature. Information is burned onto the disc by laser beam, as it is on a CD-ROM, yet unlike magneto-optical erasable discs, this data track is not susceptible to high-intensity magnetic fields. Increasingly, computers with immense storage requirements are being placed in laboratories and hospitals where such magnetic fields are commonplace.

To use WORM effectively, data must be organized to highlight the media's mass storage ►

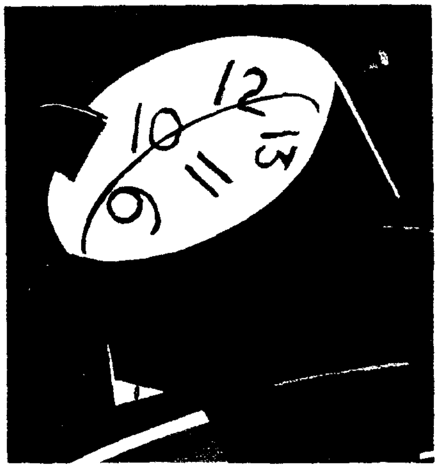
ILLUSTRATION BY MIKE McLAUGHLIN

and indelible storage characteristic while hiding or minimizing its slow seek time. A *file system* is part of the computer's operating system software that controls how information is organized on the disc.

The file system maintains directories and subdirectories, and performs mundane tasks such as reading data out of files and giving it to application programs. Most operating systems have their own proprietary file system, which explains why one operating system's floppy disks cannot readily be used with another system.

A write-once file system should make the disc appear to be a multiple-write, or erasable, disc under normal circumstances. It should let users arbitrarily create, change, delete and rename files and directories. Users should not have to know that the computer is specifically using a WORM drive for storage.

One way to create such a multiple-write illusion is to take advantage of WORM's storage capacity and record successive versions of files and directories. When a file's contents are changed, a new version is written to the disc. When a file is renamed, a new directory with the new name is recorded. The file system keeps track of which version of each file and directory is the most current.



Old Files Don't Die

The advantage of a WORM file system is that it permits access to previous versions of every file. Files can be retrieved after they've been "deleted," since deletion merely means removing the file's entry from the directory.

When the disc gets filled up, the file system should automatically copy the most recent version of every file onto a new disc. The old one is put away on a shelf as an archival backup.

Between 1985 and 1987, I was employed as an undergraduate researcher in the MIT Media Laboratory's Electronic Publishing Group, where I developed a file system that meets the above-mentioned requirements.

Tuning In To WOFS

I rewrote CDFS in August 1987, changing the implementation and the standard to take into account what was known about actual write-once devices.

The new file system is called the Write-Once File System, or WOFS.

In March 1985, we set out to design a file system for write-once discs, knowing absolutely nothing about the direction in which the technology was headed. It was important to make as few assumptions as possible, so we made only two: that successive blocks could be recorded, and that a write-once drive could tell the difference between an unwritten (or virgin) block and one that had information on it.

Because we were designing a file system from scratch, we also took the opportunity to fix many of the problems we saw in conventional file systems. We increased the amount of information stored with each file so the files could be read back on different operating systems. We also stored extraneous information that would be useful for performance analysis. Finally, we added redundant information, to allow files to be located and restored in case of media damage, and to allow automatic consistency checks of the file system.

Our model was designed so that write-once discs would be used for personal file storage: a user would be able to carry files from computer to computer. Therefore, like the High Sierra specification for CD-ROMs, our file system had to define a transportable standard—one that was usable on different operating systems and different types of computers.

Like High Sierra for WORM

WOFS is similar to High Sierra. Both specifications incorporate much information, on a file-by-file basis, usually not found in traditional file systems. Both provide for a path table (called a *directory list* in WOFS), which functions as a database of every directory on the disc. Compound file names (L:/wofs/file/outline, for example) can be translated quickly and efficiently.

High Sierra and WOFS also provide for multiple versions of files. Both are generous in the number of bytes allocated for storing binary values.

The major difference between the two is that High Sierra defines specific places on the disc where the path table and root directory must be located, while WOFS defines a technique for quickly locating the most recent version of the structures. WOFS can, therefore, handle multiple versions of directories as well as multiple versions of files—the key to using write-once media.

WOFS locates the most recent version of a directory list by finding the last block on the disc that was written. This is an easy task, because WOFS only writes successive blocks on the disc.

WOFS, therefore, avoids two traits of MS-DOS and most other personal computer operating systems: it does not leave empty blocks that it returns to and "fills in" later, nor does it grow user data up from the bottom of the disc while growing directory information down from the top.

Instead, WOFS mixes file data and directories, writing them out successively. The last block recorded is found with a binary search that compares written with unwritten blocks. With a Philips CM-100 CD-ROM drive, this search takes about four seconds.

The last block on the disc contains, among other information, the directory list's address which, in turn, contains addresses of the most recent version of every directory. Each directory contains the addresses of the most recent version

It was initially called the Compact Disc File System (CDFS), to be used with write-once CDs. While the first version of CDFS was in operation by September 1985, write-once CDs were not. Still, we realized we could employ the CD-ROM simulator, used to develop CDFS as a mastering platform, for making "write-once CD-ROMs."

During the next two years, CDFS was used to master four usable discs: a test disc, a disc of the CIA's World Databank II, a disc containing digitized frames from television broadcasts, and the *Thesaurus Linguae Graecae*, a database of the surviving works of 170 major Greek authors. The last disc was made at the Institute for Research and Scholarship and Information (IRIS) at Brown University. (See the related article on IRIS in this issue of *CD-ROM Review*.)

All discs were stamped at the 3M Optical Recording Project in Menomonie, Wisc.

While employed at IRIS during the summer of 1987, I developed a program that allowed a CDFS disc to be read through a high-speed network using Sun Microsystem's Network File System (NFS) protocol.

This system allows a single CD-ROM drive to provide files to many computers. In tests, we found the CD-ROM's performance over the network to be similar to that of a magnetic disk.

The article below describes how the new file system was developed. ●

Simson Garfinkel is a freelance writer living in Somerville, Mass. He wishes to acknowledge J. Spencer Love, formerly of MIT Information Systems, who contributed significantly to the design of the early CDFS.

of every file. As a result, the most recent version of any file can be located quickly.

An End of Transaction (EOT) block is the last written block on the disc. In a special block of information, called the file header, every file and every directory contains the address of the previous version of that file. For example, to locate the tenth version of a file that is currently in its fifteenth version, the file system scans backwards through the previous five versions.

To undelete a file, the file system scans backwards through previous versions of the specified directory until it finds a version with the file in it. The file's directory entry is then copied into the current directory, which is rewritten to the disc.

EOT, Phone Home

If the last block on the disc does not contain an EOT, WOFS can integrate a block and determine with near certainty whether the block is a file header, directory, directory list, or EOT. Each block is tagged with its own number, a self-referencing pointer, and a block type code.

Sixteen bytes must be matched exactly to identify a block as a file system block. Because of the self-referential pointer, the bytes are different for every block, making a mismatch exceedingly low.

If the last block on the disc is not an EOT, the file system scans backwards until it finds one. It then scans forwards, picking up each new file as it goes along and inserting entries for the files into the appropriate directory. WOFS then writes out the new directories, a new directory list, and a new EOT (which is now the last block on the disc).

Many common criticisms of computer-based information systems—the storage media's fragility, ease of corruption or malicious data alteration, and limited storage capacity—are less important or non-existent with write-once systems. Instead of trying to work around the "problems" of write-once storage and waiting for the day that magneto-optical drives are delivered, WOFS respects WORM's uniqueness and potential usefulness.

As write-once technology becomes more available in the coming year, developers should consider using WORM to its full advantage rather than trying to make it fit into the old molds of magnetic storage systems. ●

—Simson Garfinkel