Inter-Office Memorandum

To	Alto Programmers	Date	February 21, 1979
From	Ed Taft	Location	Palo Alto
Subject	Alto File Dates (edition 3)	Organization	PARC/CSL

XEROX

Filed on: [Maxc1]<AltoDocs>FileDates.press

This is a new standard for the handling of the date and time stamps on Alto files. It has been approved by all those people directly responsible for standard Alto software that deals with dates, and is now being implemented. At some point, the material in this memo should be incorporated into the Alto OS manual and Mesa system documentation.

Changes since the last edition: handling of Rename; conditions under which changing the creation date is not appropriate.

Motivation

File dates (in particular, the *creation* and *write* dates) have never been handled in a uniform or documented fashion. Consequently, it has never been possible to use them to make decisions mechanically (e.g., to update files automatically) with any degree of confidence.

A caveat is in order. We all know that the general problem of consistency in distributed systems is hard, and that the "right" way to attack the problem involves the use of unique IDs, as is done in Juniper and Pilot. I have no illusions about being able to do things "right" in the Alto file system. Rather, I wish simply to standardize the use of an existing mechanism so that certain properties of Alto files can be relied upon. The solution that has minimal impact on existing systems is preferable.

Interpretation of file dates under the new standard

A file's *creation date* is a stamp generated when the *information* in a file is created. When a file is copied (without modification), the creation date should be copied with it. When a file is modified in any way (either in-place or as a result of being overwritten by newly-created information), a new creation date should be generated. See below for an unusual exception to this rule.

A file's *write date* is updated whenever that file is physically written on within a given file system.

A file's *read date* is updated whenever that file is physically read from within a given file system.

Discussion

Each Alto file has three time stamps: *creation*, *write*, and *read*. The Alto OS, Mesa runtime, and other software change these stamps in various ways depending on what operations are invoked by client programs.

The following discussion attempts to motivate the interpretation of creation and write dates as presented above and to detail the operations that software must perform in order to support that interpretation. No concensus has been reached on the purpose of the read date or the uses to which it might be put; therefore, the standard simply states current practice without attempting to justify it. The read date will not be mentioned further in this memo.

The creation date is a time stamp applied at the time of the *creation of the information contained in the file*, independent of the file's location or of the number of copies that may have been made. This interpretation is substantially different from former practice (see below).

The write date simply records the time at which a particular physical file was last written upon in a particular file system, regardless of the source of the information written into the file. This is conceptually simple and conforms to former practice.

If creation and write dates are generated and interpreted uniformly in this fashion by all systems, the following operations may be performed with confidence:

1. Compare the creation dates of two versions of a given file and determine which one is more recent (i.e., contains the most recently-created information). This works even if the two versions are stored on different machines whose clocks are not in sync, so long as successive versions of the file emanate from a single source whose clock is monotonic. For example, FTP uses this comparison to control the "update" feature invoked by "Retrieve/u".

2. Compare the write date of a file stored on a particular machine with some other date derived from the same machine's clock. This is useful in deciding when to perform various auxiliary operations such as recomputing hints or making backup copies. Smalltalk uses the write date for the former purpose and IFS for the latter.

The FTP protocol permits many file attributes, including the dates, to be sent along with a file. It is up to the receiving FTP to store those dates with the file in whatever manner is appropriate. What the Alto FTP subsystem does is to set the file's creation date to that supplied by the sending FTP, but to let the write date be set to "now" by the OS. This is the correct action to perform when a file is copied verbatim (i.e., without changing the information contained within it), whether the source file is in the same file system or a different one, perhaps on a different machine.

When a file is renamed, the operation should be equivalent to copying it verbatim and then deleting the source, insofar as its effects on dates are concerned. That is, the creation date should be carried over and a new write date generated.

Comparison with former practice

Here is a description of the past handling of the creation date in the Bcpl world (the Alto OS and FTP), though I believe the corresponding Mesa facilities work similarly.

The creation date is set when a file is *created*, i.e., when a new serial number is assigned and a leader page allocated. The Alto OS also sets it when a file is *recreated* in the sense of re-using the same disk space for a new version of the file, which generally occurs only when version numbering is enabled.

Assuming file version numbering is not enabled, when a file is *overwritten* the creation date is not changed, even if the new information bears no relationship to the old.

One might inquire about the intended interpretation of creation dates as evidenced by the default handling of them by existing Alto software (and, I might add, by several other operating systems, including Tenex). The best definition I can come up with is that the creation date is the time at which a file with a particular name (or perhaps with a particular collection of disk pages assigned) first came into existence in the local file system.

While such a definition may be of some use, I think the definitions that permit operations (1) and (2) above to be performed are more valuable. Given that Alto files have only two date properties pertaining to write access, rather than three, I believe it is appropriate to change the interpretation of the creation date. It was suggested that additional file date properties be implemented, either by appending new fixed fields to the standard leader page or by making use of the "leader properties" mechanism, for which space is allocated in the leader page but which has never been used by any standard software. This suggestion is not being adopted because of its large impact on existing software.

Modifications required to existing software

The write date is already handled according to the standard. Software need be modified only to conform to the creation date standard.

First, we observe that very few programs simply *copy* files, as opposed to modifying existing files or generating new ones (perhaps derived from earlier ones, as in an editor). FTP and the Executive's "Copy" command are examples of widely-used programs that copy files, and there aren't very many others. (TFU and Neptune (aka Put) are the only other Bcpl programs that come to mind.)

Consequently, the default action provided in file access software (the Alto OS, Mesa system, etc.) should be *unconditionally to set the creation date to "now"* when a file is written on in any way. A facility should be provided whereby the client program may set the creation date to some other value. *In no case* is it appropriate to leave the creation date as it was before.

An exception arises in a few obscure instances. If a file is modified in a manner that does not affect its logical contents insofar as they are usually interpreted, the creation date should not be changed. For example, a few programs modify themselves the first time they are executed in a new environment (e.g., to cache file hints). It would be undesirable if the resulting file then appeared to be newer than the one from which it had been copied originally.

The following software has to be changed to set the creation date to "now" during writes, as just explained:

Alto Operating System

Bravo (because it doesn't use the OS file system procedures)

Mesa system

Smalltalk system

The following software has to be changed to set the destination file's creation date equal to that of the source file when a verbatim copy operation is performed:

Executive ("Copy" command)

TFU

IFS

Tenex

The FTP and Neptune subsystems already do this.

Schedule

To minimize operational problems while the software is being converted, it is necessary that *all* user software be updated before *any* of the file server software. I believe there are no other interdependencies.

The implementation schedule is tied closely to the release of Mesa 5. Current plans call for new versions of the Alto OS (version 16), Bravo (versions 7.3 and 9.?), and Executive (version 10) to be available well before the formal release of Mesa 5. At some suitable time after the Mesa 5 release, the IFS and Tenex file servers will switch to the new standard.