Ruby master - Feature #4052

File.lutime Patch

11/14/2010 10:23 AM - runpaint (Run Paint Run Run)

Status:	Closed	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:	2.6	

Description

=begin
This patch implements File.lutime, which behaves as File.utime except if given a symlink it acts upon the link itself rather than the referent. The naming convention follows that of the other singleton methods of File which treat symlinks specially. This functionality could be implemented by a gem, but given we already define File.lchmod, File.lchown, and File.lstat, it is inconsistent not to also define File.lutime.

This functionality is provided by either lutimes(3), or utimensat(2) with the AT_SYMLINK_NOFOLLOW flag. The latter is standardised by POSIX http://goo.gl/38pHx

Platform support is as follows:

- FreeBSD - Has lutimes(2) since version 3. http://goo.gl/9OWRF
- Mac OS - Has lutimes since at least version 10.5. http://goo.gl/pSvM1
- OpenBSD - Appears to lack both APIs.
- Linux - Supported lutimes(3) and utimensat(2) since version 2.6.22. (lutimes() is just a wrapper for utimensat()). http://goo.gl/p29ja
- Windows - No support.

For the status of esoteric platforms, consult http://goo.gl/Bf4Qm and http://goo.gl/7h6cj .

On platforms that don't support either function, File.lutime raises a NotImplementedError.

The portability logic is a bit hairy, so I'll summarise:

1 - File.lutime requires either lutimes() to be defined or both utimensat() and AT_SYMLINK_NOFOLLOW. I'm not aware of a practical case where utimensat() would be defined without AT_SYMLINK_NOFOLLOW. Further, File.lutime requires utimes(). This isn't strictly necessary, but simplifies the logic: it seems most unlikely that a platform would support either lutimes() or utimensat() without also supporting utimes().
2 - Like File.utime, we prefer to use utimensat() as it provides better resolution than utimes().
3 - If utimensat(..., AT_SYMLINK_NOFOLLOW) sets errno to ENOSYS, there are two possibilities: either the platform doesn't support the system call, or, like Linux 2.6.22 (http://goo.gl/Bf4Qm), utimensat() is defined but raises ENOSYS when given this flag. If utimensat() gives ENOSYS when it wasn't passed AT_SYMLINK_NOFOLLOW, we assume the syscall really doesn't exist, and bypass it for all future invocations of File.utime and File.lutime. If utimensat() gives ENOSYS when passed AT_SYMLINK_NOFOLLOW, or AT_SYMLINK_NOFOLLOW_ isn't defined, we bypass utimensat() for all future File.lutime invocations, but still consider it for future File.utime invocations.
4 - If we can't use utimensat() we fall back on lutimes().
5 - If a platform defines utimensat() and AT_SYMLINK_NOFOLLOW, but the former gives ENOSYS when passed the latter, we also fall back on lutimes(). However, if the platform doesn't define lutimes(), we raise NotImplementedError.

Tests are attached for File.utime when given a symlink and File.lutime.
=end

Associated revisions

Revision defcaf89 - 11/29/2017 09:59 AM - nobu (Nobuyoshi Nakada)

file.c: File.lutime

- file.c (utime_internal): add File.lutime. [Feature #4052]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60933 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 60933 - 11/29/2017 09:59 AM - nobu (Nobuyoshi Nakada)

file.c: File.lutime

- file.c (utime_internal): add File.lutime. [Feature #4052]

Revision 60933 - 11/29/2017 09:59 AM - nobu (Nobuyoshi Nakada)

file.c: File.lutime

- file.c (utime_internal): add File.lutime. [Feature #4052]

Revision 60933 - 11/29/2017 09:59 AM - nobu (Nobuyoshi Nakada)

file.c: File.lutime

- file.c (utime_internal): add File.lutime. [Feature #4052]

Revision 6875e324 - 11/29/2017 02:48 PM - nobu (Nobuyoshi Nakada)

configure.ac: fixed a typo [Feature #4052]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60939 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 60939 - 11/29/2017 02:48 PM - nobu (Nobuyoshi Nakada)

configure.ac: fixed a typo [Feature #4052]

Revision 60939 - 11/29/2017 02:48 PM - nobu (Nobuyoshi Nakada)

configure.ac: fixed a typo [Feature #4052]

Revision 60939 - 11/29/2017 02:48 PM - nobu (Nobuyoshi Nakada)

configure.ac: fixed a typo [Feature #4052]

History

#1 - 11/16/2010 07:16 PM - naruse (Yui NARUSE)

- Status changed from Open to Assigned

- Assignee set to matz (Yukihiro Matsumoto)

=begin
NetBSD has lutime(2).
http://netbsd.gw.com/cgi-bin/man-cgi?lutimes++NetBSD-current

Windows Vista or later can set symbolic on NTFS by CreateFile with FILE_FLAG_OPEN_REPARSE_POINT flag and SetFileTime.
http://msdn.microsoft.com/en-us/library/aa365682(VS.85).aspx Symbolic Link Effects on File Systems Functions
http://msdn.microsoft.com/en-us/library/ms724933(VS.85).aspx SetFileTime Function
http://msdn.microsoft.com/en-us/library/aa363858(VS.85).aspx CreateFile Function

=end

#2 - 05/02/2012 12:40 AM - akr (Akira Tanaka)

- Description updated

I found that File.lutimes can be used in fileutils.rb.

Recently I fixed copy_metadata method in fileutils.rb.
It copies metadata (atime, mtime, owner, group and mode) but it doesn't care symbolic links.

I changed the method to support symbolic links.
However atime and mtime could not be copied because we don't have File.lutime.

#3 - 11/20/2012 10:39 PM - mame (Yusuke Endoh)

- Target version set to 2.6

#4 - 11/29/2017 08:48 AM - matz (Yukihiro Matsumoto)

Accepted.

Matz.

#5 - 11/29/2017 09:59 AM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

Applied in changeset trunk|r60933.

---

file.c: File.lutime

- file.c (utime_internal): add File.lutime. [Feature #4052]

Files

lutime.patch	6.77 KB	11/14/2010	runpaint (Run Paint Run Run)