NAME

utimes – set file times

SYNOPSIS

```
#include <sys/types.h>
#include <sys/time.h>
int utimes(char *file, struct timeval *tvp);
```

DESCRIPTION

utimes( ) sets the access and modification times of the file named by *file*.

If *tvp* is NULL, the access and modification times are set to the current time. A process must be the owner of the file or have write permission for the file to use **utimes( )** in this manner.

If *tvp* is not NULL, it is assumed to point to an array of two **timeval** structures. The access time is set to the value of the first member, and the modification time is set to the value of the second member. Only the owner of the file or the super-user may use **utimes( )** in this manner.

In either case, the *inode-changed* time of the file is set to the current time.

utimes( ) also causes the time of the last file status change (**st_ctime**) to be updated.

RETURN VALUES

utimes( ) returns:

0 on success.

–1 on failure and sets **errno** to indicate the error.

ERRORS

utimes( ) will fail if one or more of the following are true:

EACCES	Search permission is denied for a component of the path prefix of *file*.
EACCES	The effective user ID of the process is not super-user and not the owner of the file, write permission is denied for the file, and *tvp* is NULL.
EFAULT	*file* or *tvp* points to an illegal address.
EINTR	A signal was caught during the **utimes( )** function.
EINVAL	The number of microseconds specified in one or both of the **timeval** structures pointed to by *tvp* was greater than or equal to 1,000,000 or less than 0.
EIO	An I/O error occurred while reading from or writing to the file system.
ELOOP	Too many symbolic links were encountered in translating *file*.
EMULTIHOP	Components of *file* require hopping to multiple remote machines and the file system does not allow it.
ENAMETOOLONG	The length of the *file* argument exceeds {**PATH_MAX**}, or the length of a path component of *file* exceeds {**NAME_MAX**} while {**_POSIX_NO_TRUNC**} is in effect.

ENOENT	The named file does not exist or is a null pathname.
ENOLINK	*file* points to a remote machine and the link to that machine is no longer active.
ENOTDIR	A component of the path prefix of *file* is not a directory.
EPERM	The effective user of the calling process is not super-user and not the owner of the file, and *tvp* is not **NULL**.
EROFS	The file system containing the file is mounted read-only.

SEE ALSO **stat**(2)