Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'ftw.3p' command

$ man ftw.3p

FTW(3P) POSIX Programmer's Manual FTW(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

ftw ? traverse (walk) a file tree

SYNOPSIS

```
#include <ftw.h>
int ftw(const char *path, int (*fn)(const char *,
    const struct stat *ptr, int flag), int ndirs);
```

DESCRIPTION

The ftw() function shall recursively descend the directory hierarchy rooted in path. For each object in the hierarchy, ftw() shall call the function pointed to by fn, passing it a pointer to a null-terminated character string containing the name of the object, a pointer to a stat structure containing information about the object, filled in as if stat() or lstat() had been called to retrieve the information. Possible values of the integer, defined in the <ftw.h> header, are:

FTW_D For a directory.

FTW_DNR For a directory that cannot be read.

FTW_F For a non-directory file.

FTW_SL For a symbolic link (but see also FTW_NS below).

FTW_NS For an object other than a symbolic link on which stat() could not successfully be executed. If the object is a sym? bolic link and stat() failed, it is unspecified whether ftw() passes FTW_SL or FTW_NS to the user-supplied function.

If the integer is FTW_DNR, descendants of that directory shall not be processed. If the integer is FTW_NS, the stat structure contains unde? fined values. An example of an object that would cause FTW_NS to be passed to the function pointed to by fn would be a file in a directory with read but without execute (search) permission.

The ftw() function shall visit a directory before visiting any of its descendants.

The ftw() function shall use at most one file descriptor for each level in the tree.

The argument ndirs should be in the range [1,{OPEN_MAX}].

The tree traversal shall continue until either the tree is exhausted, an invocation of fn returns a non-zero value, or some error, other than [EACCES], is detected within ftw().

The ndirs argument shall specify the maximum number of directory streams or file descriptors or both available for use by ftw() while traversing the tree. When ftw() returns it shall close any directory streams and file descriptors it uses not counting any opened by the ap? plication-supplied fn function.

The results are unspecified if the application-supplied fn function does not preserve the current working directory.

The ftw() function need not be thread-safe.

RETURN VALUE

If the tree is exhausted, ftw() shall return 0. If the function pointed to by fn returns a non-zero value, ftw() shall stop its tree traversal and return whatever value was returned by the function pointed to by fn. If ftw() detects an error, it shall return -1 and set errno to in? dicate the error.

If ftw() encounters an error other than [EACCES] (see FTW_DNR and

FTW_NS above), it shall return -1 and set errno to indicate the error.

The external variable errno may contain any error value that is possi?
ble when a directory is opened or when one of the stat functions is ex?
ecuted on a directory or file.

ERRORS

The ftw() function shall fail if:

EACCES Search permission is denied for any component of path or read permission is denied for path.

ELOOP A loop exists in symbolic links encountered during resolution of the path argument.

ENAMETOOLONG
The length of a component of a pathname is longer than {NAME_MAX}.

ENOENT A component of path does not name an existing file or path is an empty string.

ENOTDIR
A component of path names an existing file that is neither a di?
rectory nor a symbolic link to a directory.

EOVERFLOW
A field in the stat structure cannot be represented correctly in the current programming environment for one or more files found in the file hierarchy.

The ftw() function may fail if:

EINVAL The value of the ndirs argument is invalid.

ELOOP More than {SYMLOOP_MAX} symbolic links were encountered during resolution of the path argument.

ENAMETOOLONG
The length of a pathname exceeds {PATH_MAX}, or pathname resolu?
tion of a symbolic link produced an intermediate result with a length that exceeds {PATH_MAX}.

In addition, if the function pointed to by fn encounters system errors, errno may be set accordingly.

The following sections are informative.

EXAMPLES

Walking a Directory Structure

The following example walks the current directory structure, calling the fn function for every directory entry, using at most 10 file de? scriptors:

```
#include <ftw.h>
...
if (ftw(".", fn, 10) != 0) {
    perror("ftw"); exit(2);
}
```

APPLICATION USAGE

The ftw() function may allocate dynamic storage during its operation. If ftw() is forcibly terminated, such as by longjmp() or siglongjmp() being executed by the function pointed to by fn or an interrupt rou? tine, ftw() does not have a chance to free that storage, so it remains permanently allocated. A safe way to handle interrupts is to store the fact that an interrupt has occurred, and arrange to have the function pointed to by fn return a non-zero value at its next invocation.

Applications should use the nftw() function instead of the obsolescent ftw() function.

RATIONALE

None.

FUTURE DIRECTIONS

The ftw() function may be removed in a future version.

SEE ALSO

fdopendir(), fstatat(), longjmp(), nftw(), siglongjmp()

The Base Definitions volume of POSIX.1?2017, <ftw.h>, <sys_stat.h>

COPYRIGHT

event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at http://www.opengroup.org/unix/online.html .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.ker?nel.org/doc/man-pages/reporting_bugs.html .