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

tmpfile — create a temporary file

SYNOPSIS

#include <stdio.h>

FILE *tmpfile(void);

DESCRIPTION

The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1-2008 defers to the ISO C standard.

The *tmpfile*() function shall create a temporary file and open a corresponding stream. The file shall be automatically deleted when all references to the file are closed. The file is opened as in *fopen*() for update (*w*+), except that implementations may restrict the permissions, either by clearing the file mode bits or setting them to the value S_IRUSR | S_IWUSR.

In some implementations, a permanent file may be left behind if the process calling *tmpfile*() is killed while it is processing a call to *tmpfile*().

An error message may be written to standard error if the stream cannot be opened.

RETURN VALUE

Upon successful completion, *tmpfile*() shall return a pointer to the stream of the file that is created. Otherwise, it shall return a null pointer and set *errno* to indicate the error.

ERRORS

The *tmpfile*() function shall fail if:

EINTR
A signal was caught during *tmpfile*().

EMFILE
All file descriptors available to the process are currently open.

EMFILE
{STREAM_MAX} streams are currently open in the calling process.

ENFILE
The maximum allowable number of files is currently open in the system.

ENOSPC
The directory or file system which would contain the new file cannot be expanded.

EOVERFLOW
The file is a regular file and the size of the file cannot be represented correctly in an object of type **off_t**.

The *tmpfile*() function may fail if:

EMFILE
{FOPEN_MAX} streams are currently open in the calling process.

ENOMEM
Insufficient storage space is available.

The following sections are informative.

EXAMPLES

Creating a Temporary File

The following example creates a temporary file for update, and returns a pointer to a stream for the created file in the *fp* variable.

```
#include <stdio.h>
```

```
...
FILE *fp;

fp = tmpfile ();
```

APPLICATION USAGE

It should be possible to open at least {TMP_MAX} temporary files during the lifetime of the program (this limit may be shared with *tmpnam*()) and there should be no limit on the number simultaneously open other than this limit and any limit on the number of open file descriptors or streams ({OPEN_MAX}, {FOPEN_MAX}, {STREAM_MAX}).

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.5, *Standard I/O Streams*, *fopen*( ), *mkdtemp*( ), *tmpnam*( ), *unlink*( )

The Base Definitions volume of POSIX.1-2008, **<stdio.h>**

COPYRIGHT