



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

\$ man mkdtemp.3p

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

mkdtemp, mkstemp ? create a unique directory or file

SYNOPSIS

```
#include <stdlib.h>

char *mkdtemp(char *template);

int mkstemp(char *template);
```

DESCRIPTION

The `mkdtemp()` function shall create a directory with a unique name derived from `template`. The application shall ensure that the string provided in `template` is a pathname ending with at least six trailing 'X' characters. The `mkdtemp()` function shall modify the contents of `template` by replacing six or more 'X' characters at the end of the pathname with the same number of characters from the portable filename character set. The characters shall be chosen such that the resulting pathname does not duplicate the name of an existing file at the time of the call to `mkdtemp()`. The `mkdtemp()` function shall use the resulting pathname to create the new directory as if by a call to:

`mkdir(pathname, S_IRWXU)`

The `mkstemp()` function shall create a regular file with a unique name derived from `template` and return a file descriptor for the file open for reading and writing. The application shall ensure that the string provided in `template` is a pathname ending with at least six trailing 'X' characters. The `mkstemp()` function shall modify the contents of `template` by replacing six or more 'X' characters at the end of the `pathname` with the same number of characters from the portable filename character set. The characters shall be chosen such that the resulting `pathname` does not duplicate the name of an existing file at the time of the call to `mkstemp()`. The `mkstemp()` function shall use the resulting `pathname` to create the file, and obtain a file descriptor for it, as if by a call to:

`open(pathname, O_RDWR|O_CREAT|O_EXCL, S_IRUSR|S_IWUSR)`

By behaving as if the `O_EXCL` flag for `open()` is set, the function prevents any possible race condition between testing whether the file exists and opening it for use.

RETURN VALUE

Upon successful completion, the `mkdtemp()` function shall return the value of `template`. Otherwise, it shall return a null pointer and shall set `errno` to indicate the error.

Upon successful completion, the `mkstemp()` function shall return an open file descriptor. Otherwise, it shall return -1 and shall set `errno` to indicate the error.

ERRORS

The `mkdtemp()` function shall fail if:

EACCES Search permission is denied on a component of the `path` prefix, or write permission is denied on the parent directory of the `directory` to be created.

EINVAL The string pointed to by `template` does not end in "XXXXXX".

ELOOP A loop exists in symbolic links encountered during resolution of the `path` of the `directory` to be created.

EMLINK The link count of the parent `directory` would exceed `{LINK_MAX}`.

ENAMETOOLONG

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

ENOENT A component of the path prefix specified by the `template` argument does not name an existing directory.

ENOSPC The file system does not contain enough space to hold the contents of the new directory or to extend the parent directory of the new directory.

ENOTDIR

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory.

EROFS The parent directory resides on a read-only file system.

The `mkdtemp()` function may fail if:

ELOOP More than {SYMLINK_MAX} symbolic links were encountered during resolution of the path of the directory to be created.

ENAMETOOLONG

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

The error conditions for the `mkstemp()` function are defined in `open()`.

The following sections are informative.

EXAMPLES

Generating a Pathname

The following example creates a file with a 10-character name beginning with the characters "file" and opens the file for reading and writing.

The value returned as the value of `fd` is a file descriptor that identifies the file.

```
#include <stdlib.h>

...

char template[] = "/tmp/fileXXXXXX";

int fd;

fd = mkstemp(template);
```

It is possible to run out of letters.

Portable applications should pass exactly six trailing 'X's in the template and no more; implementations may treat any additional trailing 'X's as either a fixed or replaceable part of the template. To be sure of only passing six, a fixed string of at least one non-'X' character should precede the six 'X's.

Since 'X' is in the portable filename character set, some of the replacement characters can be 'X's, leaving part (or even all) of the template effectively unchanged.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

`getpid()`, `mkdir()`, `open()`, `tmpfile()`, `tmpnam()`

The Base Definitions volume of POSIX.1-2017, `<stdlib.h>`

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the 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.kernel.org/doc/man-pages/reporting_bugs.html.