

NAME

mkdir - make a directory

SYNOPSIS

```
#include <sys/types.h>
#include <sys/stat.h>

int mkdir(const char *path, mode_t mode);
```

DESCRIPTION

The *mkdir()* function creates a new directory with name *path*. The file permission bits of the new directory are initialised from *mode*. These file permission bits of the *mode* argument are modified by the process' file creation mask.

When bits in *mode* other than the file permission bits are set, the meaning of these additional bits is implementation-dependent.

The directory's user ID is set to the process' effective user ID. The directory's group ID is set to the group ID of the parent directory or to the effective group ID of the process.

The newly created directory will be an empty directory.

Upon successful completion, *mkdir()* will mark for update the *st_atime*, *st_ctime* and *st_mtime* fields of the directory. Also, the *st_ctime* and *st_mtime* fields of the directory that contains the new entry are marked for update.

RETURN VALUE

Upon successful completion, *mkdir()* returns 0. Otherwise, -1 is returned, no directory is created and *errno* is set to indicate the error.

ERRORS

The *mkdir()* function will 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.

[EEXIST]

The named file exists.

[ELOOP]

Too many symbolic links were encountered in resolving *path*.

[EMLINK]

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

[ENAMETOOLONG]

The length of the *path* argument exceeds {PATH_MAX} or a pathname component is longer than {NAME_MAX}.

[ENOENT]

A component of the path prefix specified by *path* does not name an existing directory or *path* is an empty string.

[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 is not a directory.

[EROFS]

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

The *mkdir()* function may fail if:

[ENAMETOOLONG]

Pathname resolution of a symbolic link produced an intermediate result whose length exceeds {PATH_MAX}.

EXAMPLES

None.

APPLICATION USAGE

None.

SEE ALSO

[*umask\(\)*](#), [*<sys/stat.h>*](#), [*<sys/types.h>*](#).

DERIVATION

Derived from the POSIX.1-1988 standard.

UNIX ® is a registered Trademark of The Open Group.

Copyright © 1997 The Open Group

[[Main Index](#) | [XSH](#) | [XCU](#) | [XBD](#) | [XCURSES](#) | [XNS](#)]
