NAME

mkdir – make a directory

SYNOPSIS

```
#include <sys/types.h>
#include <sys/stat.h>
int mkdir(const char *path, mode_t mode);
```

MT-LEVEL

Async-Signal-Safe

DESCRIPTION

mkdir( ) creates a new directory named by the path name pointed to by *path* . The mode of the new directory is initialized from *mode* (see **chmod**(2) for values of mode). The protection part of the *mode* argument is modified by the process's file creation mask (see **umask**(2)).

The directory's owner ID is set to the process's effective user ID. The directory's group ID is set to the process's effective group ID, or if the **S_ISGID** bit is set in the parent directory, then the group ID of the directory is inherited from the parent. The **S_ISGID** bit of the new directory is inherited from the parent directory.

If *path* is a symbolic link, it is not followed.

The newly created directory is empty with the exception of entries for itself (**.**) and its parent directory (**..**).

Upon successful completion, **mkdir( )** marks 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 VALUES

Upon successful completion, a value of 0 is returned. Otherwise, a value of –1 is returned, and **errno** is set to indicate the error.

ERRORS

mkdir( ) fails and creates no directory if one or more of the following are true:

EACCES	Either a component of the path prefix denies search permission or write permission is denied on the parent directory of the directory to be created.
EEXIST	The named file already exists.
EFAULT	*path* points to an illegal address.
EIO	An I/O error has occurred while accessing the file system.
ELOOP	Too many symbolic links were encountered in translating *path*.
EMLINK	The maximum number of links to the parent directory would be exceeded.
EMULTIHOP	Components of *path* require hopping to multiple remote machines and the file system type does not allow it.
ENAMETOOLONG	The length of the *path* argument exceeds {**PATH_MAX**}, or the length of a *path* component exceeds {**NAME_MAX**} while {**_POSIX_NO_TRUNC**} is in effect.

ENOENT	A component of the path prefix does not exist or is a null pathname.
ENOLINK	*path* points to a remote machine and the link to that machine is no longer active.
ENOSPC	No free space is available on the device containing the directory.
ENOTDIR	A component of the path prefix is not a directory.
EROFS	The path prefix resides on a read-only file system.

SEE ALSO **chmod**(2), **mknod**(2), **umask**(2), **stat**(5)