NAME

creat – create a new file or rewrite an existing one

SYNOPSIS

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

int creat(const char **path***, mode_t** *mode***);**

MT-LEVEL

Async-Signal-Safe

DESCRIPTION

creat( ) creates a new ordinary file or prepares to rewrite an existing file named by the path name pointed to by *path*.

If the file exists, the length is truncated to 0 and the mode and owner are unchanged.

If the file does not exist the file's owner ID is set to the effective user ID of the process. The group ID of the file is set to the effective group ID of the process, or if the **S_ISGID** bit is set in the parent directory then the group ID of the file is inherited from the parent directory. The access permission bits of the file mode are set to the value of *mode* modified as follows:

- If the group ID of the new file does not match the effective group ID or one of the supplementary group IDs, the **S_ISGID** bit is cleared.
- All bits set in the process's file mode creation mask are cleared (see **umask**(2)).
- The "save text image after execution bit" of the mode is cleared (see **chmod**(2) for the values of mode).

Upon successful completion, a write-only file descriptor is returned and the file is open for writing, even if the mode does not permit writing. The file pointer is set to the beginning of the file. The file descriptor is set to remain open across **exec** functions (see **fcntl**(2)). A new file may be created with a mode that forbids writing.

The call **creat(***path***,** *mode***)** is equivalent to:

open(*path***, O_WRONLY | O_CREAT | O_TRUNC,** *mode***)**

RETURN VALUES

Upon successful completion a non-negative integer, namely the lowest numbered unused file descriptor, is returned. Otherwise, a value of –1 is returned, no files are created or modified, and **errno** is set to indicate the error.

ERRORS

creat( ) fails if one or more of the following are true:

EACCES	Search permission is denied on a component of the path prefix.
EACCES	The file does not exist and the directory in which the file is to be created does not permit writing.
EACCES	The file exists and write permission is denied.
EAGAIN	The file exists, mandatory file/record locking is set, and there are outstanding record locks on the file (see **chmod**(2)).
EFAULT	*path* points to an illegal address.

EINTR	A signal was caught during the **creat( )** function.
EISDIR	The named file is an existing directory.
ELOOP	Too many symbolic links were encountered in translating *path*.
EMFILE	The process has too many open files (see **getrlimit**(2)).
EMULTIHOP	Components of *path* require hopping to multiple remote machines.
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.
ENFILE	The system file table is full.
ENOENT	A component of the path prefix does not exist.
ENOENT	The path name is null.
ENOLINK	*path* points to a remote machine and the link to that machine is no longer active.
ENOSPC	The file system is out of inodes.
ENOTDIR	A component of the path prefix is not a directory.
EROFS	The named file resides or would reside on a read-only file system.

SEE ALSO

chmod(2), **close**(2), **dup**(2), **fcntl**(2), **getrlimit**(2), **lseek**(2), **open**(2), **read**(2), **umask**(2), **write**(2), **stat**(5)