NAME

access – determine accessibility of a file

SYNOPSIS

#include <unistd.h>

int access(const char **path***, int** *amode***);**

MT-LEVEL

Async-Signal-Safe

DESCRIPTION

access( ) checks the file pointed to by *path* for accessibility according to the bit pattern contained in *amode*, using the real user ID in place of the effective user ID and the real group ID in place of the effective group ID. This allows a setuid process to verify that the user running it would have had permission to access this file. The bit pattern contained in *amode* is constructed by an OR of the access permissions to be checked (R_OK, W_OK, and X_OK, or the existence test, (F_OK). These constants are defined in **<unistd.h>** as follows:

R_OK Test for read permission.
W_OK Test for write permission.
X_OK Test for execute or search permission.
F_OK Check existence of file

See **intro**(2) for additional information about "File Access Permission".

RETURN VALUES

If the requested access is permitted, a value of 0 is returned. Otherwise, a value of –1 is returned and **errno** is set to indicate the error.

ERRORS

Access to the file is denied if one or more of the following are true:

EACCES	Search permission is denied on a component of the path prefix.
EACCES	Permission bits of the file mode do not permit the requested access.
EFAULT	*path* points to an illegal address.
EINTR	A signal was caught during the **access( )** function.
ELOOP	Too many symbolic links were encountered in translating *path*.
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.
ENOENT	The *path* argument points to an empty string or to the name of a file that does not exist.
ENOLINK	*path* points to a remote machine and the link to that machine is no longer active.

ENOTDIR	A component of the path prefix is not a directory.
EROFS	Write access is requested for a file on a read-only file system.

SEE ALSO **intro**(2), **chmod**(2), **stat**(2)