

NAME

`errno` - number of last error

SYNOPSIS

```
#include <errno.h>
```

DESCRIPTION

The `<errno.h>` header file defines the integer variable `errno`, which is set by system calls and some library functions in the event of an error to indicate what went wrong. Its value is significant only when the return value of the call indicated an error (i.e., -1 from most system calls; -1 or NULL from most library functions); a function that succeeds *is* allowed to change `errno`.

Valid error numbers are all nonzero; `errno` is never set to zero by any system call or library function.

For some system calls and library functions (e.g., [getpriority\(2\)](#)), -1 is a valid return on success. In such cases, a successful return can be distinguished from an error return by setting `errno` to zero before the call, and then, if the call returns a status that indicates that an error may have occurred, checking to see if `errno` has a nonzero value.

`errno` is defined by the ISO C standard to be a modifiable lvalue of type `int`, and must not be explicitly declared; `errno` may be a macro. `errno` is thread-local; setting it in one thread does not affect its value in any other thread.

All the error names specified by POSIX.1 must have distinct values, with the exception of **EAGAIN** and **EWOULDBLOCK**, which may be the same.

Below is a list of the symbolic error names that are defined on Linux. Some of these are marked *POSIX.1*, indicating that the name is defined by POSIX.1-2001, or *C99*, indicating that the name is defined by C99.

E2BIG	Argument list too long (POSIX.1)
EACCES	Permission denied (POSIX.1)
EADDRINUSE	Address already in use (POSIX.1)
EADDRNOTAVAIL	Address not available (POSIX.1)
EAFNOSUPPORT	Address family not supported (POSIX.1)
EAGAIN	Resource temporarily unavailable (may be the same value as EWOULDBLOCK) (POSIX.1)
EALREADY	Connection already in progress (POSIX.1)
EBADE	Invalid exchange
EBADF	Bad file descriptor (POSIX.1)
EBADFD	File descriptor in bad state
EBADMSG	Bad message (POSIX.1)
EBADR	Invalid request descriptor
EBADRQC	Invalid request code
EBADSLT	Invalid slot
EBUSY	Device or resource busy (POSIX.1)
ECANCELED	Operation canceled (POSIX.1)

ECHILD	No child processes (POSIX.1)
ECHRNG	Channel number out of range
ECOMM	Communication error on send
ECONNABORTED	Connection aborted (POSIX.1)
ECONNREFUSED	Connection refused (POSIX.1)
ECONNRESET	Connection reset (POSIX.1)
EDEADLK	Resource deadlock avoided (POSIX.1)
EDEADLOCK	Synonym for EDEADLK
EDESTADDRREQ	Destination address required (POSIX.1)
EDOM	Mathematics argument out of domain of function (POSIX.1, C99)
EDQUOT	Disk quota exceeded (POSIX.1)
EEXIST	File exists (POSIX.1)
EFAULT	Bad address (POSIX.1)
EFBIG	File too large (POSIX.1)
EHOSTDOWN	Host is down
EHOSTUNREACH	Host is unreachable (POSIX.1)
EIDRM	Identifier removed (POSIX.1)
EILSEQ	Illegal byte sequence (POSIX.1, C99)
EINPROGRESS	Operation in progress (POSIX.1)
EINTR	Interrupted function call (POSIX.1); see signal(7) .
EINVAL	Invalid argument (POSIX.1)
EIO	Input/output error (POSIX.1)
EISCONN	Socket is connected (POSIX.1)
EISDIR	Is a directory (POSIX.1)
EISNAM	Is a named type file
EKEYEXPIRED	Key has expired
EKEYREJECTED	Key was rejected by service
EKEYREVOKED	Key has been revoked
EL2HLT	Level 2 halted
EL2NSYNC	Level 2 not synchronized
EL3HLT	Level 3 halted

EL3RST	Level 3 halted
ELIBACC	Cannot access a needed shared library
ELIBBAD	Accessing a corrupted shared library
ELIBMAX	Attempting to link in too many shared libraries
ELIBSCN	lib section in a.out corrupted
ELIBEXEC	Cannot exec a shared library directly
ELOOP	Too many levels of symbolic links (POSIX.1)
EMEDIUMTYPE	Wrong medium type
EMFILE	Too many open files (POSIX.1)
EMLINK	Too many links (POSIX.1)
EMSGSIZE	Message too long (POSIX.1)
EMULTIHOP	Multihop attempted (POSIX.1)
ENAMETOOLONG	Filename too long (POSIX.1)
ENETDOWN	Network is down (POSIX.1)
ENETRESET	Connection aborted by network (POSIX.1)
ENETUNREACH	Network unreachable (POSIX.1)
ENFILE	Too many open files in system (POSIX.1)
ENOBUFS	No buffer space available (POSIX.1 (XSI STREAMS option))
ENODATA	No message is available on the STREAM head read queue (POSIX.1)
ENODEV	No such device (POSIX.1)
ENOENT	No such file or directory (POSIX.1)
ENOEXEC	Exec format error (POSIX.1)
ENOKEY	Required key not available
ENOLCK	No locks available (POSIX.1)
ENOLINK	Link has been severed (POSIX.1)
ENOMEDIUM	No medium found
ENOMEM	Not enough space (POSIX.1)
ENOMSG	No message of the desired type (POSIX.1)
ENONET	Machine is not on the network
ENOPKG	Package not installed
ENOPROTOPT	Protocol not available (POSIX.1)
ENOSPC	No space left on device (POSIX.1)
ENOSR	No STREAM resources (POSIX.1 (XSI STREAMS option))
ENOSTR	Not a STREAM (POSIX.1 (XSI STREAMS option))
ENOSYS	Function not implemented (POSIX.1)

ENOTBLK	Block device required
ENOTCONN	The socket is not connected (POSIX.1)
ENOTDIR	Not a directory (POSIX.1)
ENOTEMPTY	Directory not empty (POSIX.1)
ENOTSOCK	Not a socket (POSIX.1)
ENOTSUP	Operation not supported (POSIX.1)
ENOTTY	Inappropriate I/O control operation (POSIX.1)
ENOTUNIQ	Name not unique on network
ENXIO	No such device or address (POSIX.1)
EOPNOTSUPP	Operation not supported on socket (POSIX.1) (ENOTSUP and EOPNOTSUPP have the same value on Linux, but according to POSIX.1 these error values should be distinct.)
EOVERFLOW	Value too large to be stored in data type (POSIX.1)
EPERM	Operation not permitted (POSIX.1)
EPFNOSUPPORT	Protocol family not supported
EPIPE	Broken pipe (POSIX.1)
EPROTO	Protocol error (POSIX.1)
EPROTONOSUPPORT	Protocol not supported (POSIX.1)
EPROTOTYPE	Protocol wrong type for socket (POSIX.1)
ERANGE	Result too large (POSIX.1, C99)
EREMCHG	Remote address changed
EREMOTE	Object is remote
EREMOTEIO	Remote I/O error
ERESTART	Interrupted system call should be restarted
EROFS	Read-only filesystem (POSIX.1)
ESHUTDOWN	Cannot send after transport endpoint shutdown
ESPIPE	Invalid seek (POSIX.1)
ESOCKTNOSUPPORT	Socket type not supported
ESRCH	No such process (POSIX.1)
ESTALE	Stale file handle (POSIX.1) This error can occur for NFS and for other filesystems
ESTRPIPE	Streams pipe error
ETIME	Timer expired (POSIX.1 (XSI STREAMS option))

	(POSIX.1 says STREAM ioctl(2) timeout)
ETIMEDOUT	Connection timed out (POSIX.1)
ETXTBSY	Text file busy (POSIX.1)
EUCLEAN	Structure needs cleaning
EUNATCH	Protocol driver not attached
EUSERS	Too many users
EWouldBlock	Operation would block (may be same value as EAGAIN) (POSIX.1)
EXDEV	Improper link (POSIX.1)
EXFULL	Exchange full

NOTES

A common mistake is to do

```
if (somecall() == -1) {
    printf(somecall() failed\n);
    if (errno == ...) { ... }
}
```

where *errno* no longer needs to have the value it had upon return from *somecall()* (i.e., it may have been changed by the [printf\(3\)](#)). If the value of *errno* should be preserved across a library call, it must be saved:

```
if (somecall() == -1) {
    int errsv = errno;
    printf(somecall() failed\n);
    if (errsv == ...) { ... }
}
```

It was common in traditional C to declare *errno* manually (i.e., *extern int errno*) instead of including *<errno.h>*. **Do not do this.** It will not work with modern versions of the C library. However, on (very) old UNIX systems, there may be no *<errno.h>* and the declaration is needed.

SEE ALSO

[errno\(1\)](#), [err\(3\)](#), [error\(3\)](#), [perror\(3\)](#), [strerror\(3\)](#)

COLOPHON

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <http://www.kernel.org/doc/man-pages/>.