

**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.

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

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	Key has been revoked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

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

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

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