

# readlink() — Read the Value of a Symbolic Link

## Standards

Standards / Extensions	C or C++	Dependencies
POSIX.1a XPG4.2 Single UNIX Specification, Version 3	Both	

## Format

```
#define _POSIX1_SOURCE 2
#include <unistd.h>

int readlink(const char *path, char *buf, size_t bufsiz);

#define _POSIX_C_SOURCE 200112L
#include <unistd.h>

ssize_t readlink(const char *__restrict path,
                 char *__restrict buf, size_t bufsiz);
```

## General Description

Places the contents of the symbolic link *path* in the buffer *buf*. The size of the buffer is set by *bufsiz*. The result stored in *buf* does not include a terminating NULL character.

If the buffer is too small to contain the value of the symbolic link, that value is truncated to the size of the buffer (*bufsiz*). If the value returned is the size of the buffer, use `lstat()` to determine the actual size of the symbolic link.

## Returned Value

If successful, when *bufsiz* is greater than 0, `readlink()` returns the number of bytes placed in the buffer. When *bufsiz* is 0 and `readlink()` completes successfully, it returns the number of bytes contained in the symbolic link and the buffer is not changed.

If the returned value is equal to *bufsiz*, you can determine the contents of the symbolic link with either `lstat()` or `readlink()`, with a 0 value for *bufsiz*.

If unsuccessful, `readlink()` returns -1 and sets `errno` to one of the following values:

## Error Code

	Description
EACCES	Search permission is denied for a component of the path prefix.
EINVAL	The named file is not a symbolic link.
EIO	<b>Added for XPG4.2:</b> An I/O error occurred while reading from the file system.
ELOOP	A loop exists in symbolic links. This error is issued if more than <code>POSIX_SYMLINK_MAX</code> symbolic links are encountered during resolution of the <i>path</i> argument.
ENAMETOOLONG	<i>pathname</i> is longer than <b>PATH_MAX</b> characters, or some component of <i>pathname</i> is longer than <b>NAME_MAX</b> characters while <code>_POSIX_NO_TRUNC</code> is in effect. For symbolic links, the length of the pathname string substituted for a symbolic link exceeds <b>PATH_MAX</b> . The <b>PATH_MAX</b> and <b>NAME_MAX</b> values can be determined using <code>pathconf()</code> .
ENOENT	The named file does not exist.
ENOTDIR	A component of the path prefix is not a directory.