

NAME

readlinkat – read value of a symbolic link relative to a directory file descriptor

SYNOPSIS

```
#define _ATFILE_SOURCE
#include <fcntl.h> /* Definition of AT_* constants */
#include <unistd.h>
```

```
int readlinkat(int dirfd, const char *pathname,
               char *buf, size_t bufsiz);
```

DESCRIPTION

The **readlinkat()** system call operates in exactly the same way as **readlink(2)**, except for the differences described in this manual page.

If the *pathname* given in *pathname* is relative, then it is interpreted relative to the directory referred to by the file descriptor *dirfd* (rather than relative to the current working directory of the calling process, as is done by **readlink(2)** for a relative *pathname*).

If *pathname* is relative and *dirfd* is the special value **AT_FDCWD**, then *pathname* is interpreted relative to the current working directory of the calling process (like **readlink(2)**).

If *pathname* is absolute, then *dirfd* is ignored.

RETURN VALUE

On success, **readlinkat()** returns the number of bytes placed in *buf*. On error, **-1** is returned and *errno* is set to indicate the error.

ERRORS

The same errors that occur for **readlink(2)** can also occur for **readlinkat()**. The following additional errors can occur for **readlinkat()**:

EBADF

dirfd is not a valid file descriptor.

ENOTDIR

pathname is relative and *dirfd* is a file descriptor referring to a file other than a directory.

VERSIONS

readlinkat() was added to Linux in kernel 2.6.16.

CONFORMING TO

POSIX.1-2008.

NOTES

See **openat(2)** for an explanation of the need for **readlinkat()**.

SEE ALSO

openat(2), **readlink(2)**, **path_resolution(7)**

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at <http://www.kernel.org/doc/man-pages/>.