

**NAME**

**symlinkat** – create a symbolic link relative to a directory file descriptor

**SYNOPSIS**

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

```
int symlinkat(const char *oldpath, int newdirfd, const char *newpath);
```

**DESCRIPTION**

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

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

If *newpath* is relative and *newdirfd* is the special value **AT\_FDCWD**, then *newpath* is interpreted relative to the current working directory of the calling process (like **symlink(2)**).

If *newpath* is absolute, then *newdirfd* is ignored.

**RETURN VALUE**

On success, **symlinkat()** returns 0. On error, **-1** is returned and *errno* is set to indicate the error.

**ERRORS**

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

**EBADF**

*newdirfd* is not a valid file descriptor.

**ENOTDIR**

*newpath* is relative and *newdirfd* is a file descriptor referring to a file other than a directory.

**VERSIONS**

**symlinkat()** 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 **symlinkat()**.

**SEE ALSO**

**openat(2)**, **symlink(2)**, **path\_resolution(7)**, **symlink(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/>.