NAME

readlink — get symbolic link target

SYNOPSIS

readlink [**-n**|**-z**] *file*...
readlink **-f**|**-e**|**-m** *file*...

DESCRIPTION

Writes the target (content) of each *file*, which must be a symbolic link, followed by a new-line (**0xA**), to the standard output stream.

With **-fem**, canonicalises the path to *file*s by replacing itself with realpath(1).

OPTIONS

-n, **--no-newline** Write nothing after the target of each *file*.
-z, **--zero** Write a NUL after the target of each *file*.

-f, **--canonicalize** Replace with **realpath** [**-z**] *file*...
-e, **--canonicalize-existing** Replace with **realpath -e** [**-z**] *file*...
-m, **--canonicalize-missing** Replace with **realpath -m** [**-z**] *file*...

EXIT STATUS

1 if a *file* wasn't a symbolic link. An error is also issued to the standard error stream in this case.

EXAMPLES

```
$ ls -l store
lrwxrwxrwx 1 root root 25 2020-03-16  store -> /mnt/filling/store/cicada
$ readlink store
/mnt/filling/store/cicada
```

SEE ALSO

ln(1), ls(1), realpath(1), readlink(2), symlink(2)

STANDARDS

Conforms to IEEE Std 1003.1-2024 ("POSIX.1"). **-zfem** and more than one *file* are extensions, compatible with the GNU system.

HISTORY

Appeared fully-formed in OpenBSD 2.2 (1997-12) (appears in time for OpenBSD 2.1, saying it targets OpenBSD 2.1, but is not *tagged* for inclusion in OpenBSD 2.1, and is not included in OpenBSD 2.1 distribution tarballs; the first tag is for OpenBSD 2.2 and it features in OpenBSD 2.2 tarballs) as readlink(1) ("display target of symbolic link on standard output"), synopsised as

readlink [**-fn**] *file*

-n and successful behaviour is as present-day, but errors aren't:

If readlink is invoked with an argument other than the pathname of a symbolic link it exits with a non-zero exit code and without printing anything.

-f replaces readlink(2) with realpath(3); which is like this implementation's **readlink -e** (**realpath -e**). A realpath(3) failure dumps uninitialised data and a newline (unless **-n**). The result of **-f** is truncated at the length of *file*; this is fixed in OpenBSD 2.4 (1998-12).

OpenBSD 4.1 (2007-05) properly exits with an error if realpath(3) failed.

NetBSD 2.0 (2004-12) adds **readlink** to stat(1) ("display file status"), "for similarity to OpenBSD", synopsising it as

readlink [**-n**] [*file ...*]

which is like OpenBSD **readlink**, but for any number of *file*s.

NetBSD 4.0 (2007-12) adds **-f**, same as OpenBSD.

NetBSD 6.0 (2012-10) sees a **SYNOPSIS** of

readlink [**-fnqsv**] [*file . . .*]

-qs is the default (errors aren't logged) and **-v** reverses this (errors to the standard error stream).

NetBSD 10 splits the manuals into `stat`(1)/`readlink`(1).

coreutils 4.5.5 (2003-02) adds `readlink`(1) ("display value of a symbolic link"), synopsised as:

readlink [*OPTION*] *FILE*

where *OPTIONS* is actually any of **-fnqsv** 😉 This is described as equivalent to NetBSD 10; it's unclear if **-f** is as present-day or as NetBSD 10.

coreutils 5.3.0 (2005-01) adds **-em** (and notes correcting **-f**), making them as present-day.

coreutils 8.21 (2013-02) "now supports multiple arguments" and adds **-z**. **-n** is ignored with multiple *file*s.