NAME

realpath — get canonical file path

SYNOPSIS

realpath [**-e**|**-E**|**-m**|**-s**] [**-P**|**-L**] [**-z**] *file*...

DESCRIPTION

Finds the canonical path (one which starts at the root (/), has only single /es, and contains no **..** nor **.** elements) to each *file*, followed by a new-line (**0xA**), to the standard output stream; these are determined with varying degrees of laxness:

with **-e**:	*file* must exist, and its absolute path (realpath(3)) is written,
by default (**-E**):	otherwise its parent directory must exist, its (potential) absolute path is written,
with **-m**:	any prefix of *file* must exist, and the nonexistent bits are pasted at the end, then **..** and **.** elements are processed with string substitutions (like cd(1) **-L**).
With **-s**:	*file* is processed with string substitutions only.

OPTIONS

-e, **--canonicalize-existing** *file* (every component of *file*) must exist.

-E Every component but the last must exist (*file* or *file*'s parent directory). If *file* doesn't exist, the nonexistent basename is appended verbatim. This is the default.

-m, **--canonicalize-missing** No component needs to exist, longest existing prefix of *file* is canonicalised as-if with **-e**, then the nonexistent bit is pasted on the end, and **-L** processing applied.

-s, **--strip**, **--no-symlinks** Apply **-L** processing only. The filesystem is not consulted at all.

-P, **--physical** Do not pre-process *file* (resolve it as-if given to cd(1) **-P** or open(2)). This is the default.

-L, **--logical** Pre-process each *file* as-if given to cd(1) **-L**, flattening away **..** and **.**s: a/b/./c/../d turns into a/b/d, regardless of what c is (this is a string operation, the filesystem is not consulted), before giving it to **-eEms**.

-z, **--zero** Write a NUL after the canonical path of each *file* instead.

EXIT STATUS

1 if a *file* (**-e**) or its parent (**-eE**) don't exist. An error is also issued to the standard error stream in this case.

EXAMPLES

```
$ ls -l src
lrwxrwxrwx 1 root root 14 2020-04-15  src -> /usr/local/src
$ realpath src
/usr/local/src
$ realpath src/..
/usr/local
$ realpath -L src/..
/home/cicada

$ ls
ksmctl.c
$ realpath . ksmctl.c rp-pppoe one/piece/..
/usr/local/src
/usr/local/src/ksmctl.c
/usr/local/src/rp-pppoe
realpath: one/piece/..: No such file or directory
```

```
$ realpath -e . ksmctl.c rp-pppoe one/piece/..
/usr/local/src
/usr/local/src/ksmctl.c
realpath: rp-pppoe: No such file or directory
realpath: one/piece/..: No such file or directory
$ realpath -m . ksmctl.c rp-pppoe one/piece/..
/usr/local/src
/usr/local/src/ksmctl.c
/usr/local/src/rp-pppoe
/usr/local/src/one
```

SEE ALSO

`ls`(1), `pwd`(1), `readlink`(1), `symlink`(2), `realpath`(3)

STANDARDS

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

The standard mandates **-EP** behaviour only if **-E** is specified. Many implementations default to some combination of **-e** and **-L**.

HISTORY

Appeared in FreeBSD 4.3 (2001-04) and FreeBSD 5.0 (2003-01) (originally slated for FreeBSD 5.0 and appears in the FreeBSD 5.0 tag, but the manual was later changed to say it originated in even though it’s not in the FreeBSD 4.3 tag, but it *is* in the releases) as `realpath`(1) ("return resolved physical path"), synopsised as

realpath *path*

It’s actually the same binary as **pwd**: it writes `getcwd`(3) if invoked with no arguments, and `realpath`(3) if with one. This is equivalent to having a default *file* of ".". This corresponds to present-day **-e**.

FreeBSD 5.1 (2003-07) unlinks **realpath** from **pwd** and removes the default-"." behaviour.

FreeBSD 8 (2009-11) sees a **SYNOPSIS** of
realpath [**-q**] *path* [*. . .*]
-q is equivalent to **2>**`/dev/null`.

FreeBSD 9 (2012-01) sees a **SYNOPSIS** of
realpath [**-q**] [*path . . .*]
Yes, the default *file . . .* is, ".".

realpath `mksh`(1) R35 (2008-07) ships **realpath** as a built-in (used when no flags are given), writing `realpath`(3) directly (**-e**-equivalent behaviour).

`mksh`(1) R39b (2010-01) changes it to behave more like this implementation’s **-EL** (functionally this may actually be closer to **-m**).

OpenBSD 7.1 (2022-04) includes `realpath`(1) ("print the canonicalized absolute pathname"), with a **SYNOPSIS** of
realpath [**-q**] *file*
writing `realpath`(3) directly (**-e**-equivalent behaviour).

NetBSD 10 (2024-03) includes a IEEE Std 1003.1-2024 (“POSIX.1”)-compatible `readlink`(1) ("return resolved canonical path"), synopsised as
realpath [**-eEq**] [*path . . .*]
with the default *path* being "." and the default behaviour being **-E**.

coreutils 8.15 (2012-01) adds `realpath`(1) ("print the resolved"), synopsised as:

realpath [*OPTION*]... *FILE*...
where *OPTIONS* is actually any of **-emsPLz**, as present-day, and **-q**, **--relative-to**=*FILE*, **--relative-base**=*FILE* 🙄 .