NAME

resolvepath(), resolvenpath(), resolvefpath() - resolve all symbolic links of a path name

SYNOPSIS

```
#include <schily/schily.h>

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

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

int
resolvefpath(path, buf, bufsiz, flags)
        const char    *path;
              char    *buf;
              size_t  bufsiz;
              int     flags;
```

DESCRIPTION

resolvepath() takes a relative path name and resolves all symbolic links in the path name. The result is a path name that is free of symbolic links. *path* is relative path name that is used as the input. *buf* is the buffer used for the result of the conversion. *bufsiz* is the size of the result buffer.

All ‘‘.’’ components are eliminated and every non-leading ‘‘..’’ component is eliminated together with its preceding directory component. If leading ‘‘..’’ components reach to the root directory, they are replaced by ‘‘/’’.

resolvenpath() behaves like **resolvepath**() except that the file does not need to exist.

resolvefpath() takes an additional *flags* parameter from the set of flags from the following set:

RSPF_EXIST All path components must exist.

RSPF_NOFOLLOW_LAST
Don’t follow symbolic links in the last path component.

RETURN VALUE

Upon successful completion, **resolvepath**(), **resolvenpath**() and **resolvefpath**() return the number of bytes placed in the buffer Otherwise, -1 is returned and errno is set to indicate the error. In case of an error, the contents of result buffer is left in an intermediate state.

ERRORS

ERANGE The path does not fit into the suplied buffer.

EFAULT A null pointer was suplied as pathname.

EINVAL An empty relative path was supplied.

other Any other value that may be a result of an underlying filesystem operation.

USAGE

Applications should not assume that the returned contents of the buffer are null-terminated. This is because the function **resolvepath**(3) may be the Solaris version if the program is compiled on Solaris.

SEE ALSO

resolvepath(3), **resolvenpath**(3), **resolvefpath**(3)

NOTES

none