

NAME

undelete - attempt to recover a deleted file

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <unistd.h>
```

int

```
undelete(const char *path);
```

DESCRIPTION

The **undelete()** system call attempts to recover the deleted file named by *path*. Currently, this works only when the named object is a whiteout in a union file system. The system call removes the whiteout causing any objects in a lower layer of the union stack to become visible once more.

Eventually, the **undelete()** functionality may be expanded to other file systems able to recover deleted files such as the log-structured file system.

RETURN VALUES

The **undelete()** function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **undelete()** succeeds unless:

[ENOTDIR] A component of the path prefix is not a directory.

[ENAMETOOLONG] A component of a pathname exceeded 255 characters, or an entire path name exceeded 1023 characters.

[EEXIST] The path does not reference a whiteout.

[ENOENT] The named whiteout does not exist.

[EACCES] Search permission is denied for a component of the path prefix.

[EACCES] Write permission is denied on the directory containing the name to be undeleted.

- [ELOOP] Too many symbolic links were encountered in translating the pathname.
- [EPERM] The directory containing the name is marked sticky, and the containing directory is not owned by the effective user ID.
- [EINVAL] The last component of the path is ‘.’.
- [EIO] An I/O error occurred while updating the directory entry.
- [EINTEGRITY] Corrupted data was detected while reading from the file system.
- [EROFS] The name resides on a read-only file system.
- [EFAULT] The *path* argument points outside the process’s allocated address space.

SEE ALSO

unlink(2), mount_unionfs(8)

HISTORY

The **undelete**() system call first appeared in 4.4BSD-Lite.