

## NAME

readdirplus - read a directory and return attributes

## SYNOPSIS

```
#include <sys/types.h>
#include <sys/stat.h>
#include <dirent.h>
```

```
struct dirent_plus *readdirplus(DIR *dirp);
int readdirplus_r(DIR *dirp, struct dirent_plus *entry,
    struct dirent_plus **result);
```

## DESCRIPTION

The `readdirplus()` and `readdirplus_r()` functions return information about a directory entry. The results are returned in the form of a `struct dirent_plus` structure, which contains the following fields:

```
struct dirent_plus {
    struct dirent    d_dirent; /* dirent struct for this entry */
    struct stat      d_stat;   /* attributes for this entry */
    int              d_stat_err;
};
```

The `readdirplus()` function returns a pointer to a structure representing the directory entry and its attributes at the current position in the directory stream specified by the argument `dirp`, and positions the directory stream at the next entry. It returns a null pointer upon reaching the end of the directory stream. The structure `dirent` defined by the `<dirent.h>` header describes a directory entry, and the structure `stat` defined by the `<sys/stat.h>` header describes the attributes of the directory entry.

The `d_dirent` field is filled in as if by `readdir()` or `readdir_r()`, and the `d_stat` field is filled in as if by `lstat()`. Since this is a combination operation, it is possible that the `readdir()` portion of the operation succeeds, but the `lstat()` portion fails. In the event that the `lstat()` operation fails, the `d_stat_err` field will be populated with the error number that corresponds to the `lstat()` failure. If the `lstat()` operation succeeds, the `d_stat_err` field will be 0 and the `d_stat` field will contain the attributes of the directory entry.

The pointer returned by `readdirplus()` points to data which may be overwritten by another call to `readdirplus()` on the same directory stream. This data is not overwritten by another call to `readdirplus()` on a different directory stream. The `readdirplus_r()` function operates identically to the `readdirplus()` function, except that the result is stored in caller-provided memory, which will not

be overwritten by another call to `readdirplus_r()`, unless the caller passes in a pointer to the same memory.

The `readdirplus()` and `readdirplus_r()` functions will not return directory entries containing empty names. It is not specified whether they return entries for "." and "..".

If a file is removed from or added to the directory after the most recent call to `opendir()` or `rewinddir()`, whether a subsequent call to `readdirplus()` or `readdirplus_r()` returns an entry for that file is unspecified.

#### RETURN VALUE

On successful completion, `readdirplus()` returns a pointer to an object of type `struct dirent_plus`. When an error is encountered, the value `NULL` is returned and `errno` is set to indicate the error. When the end of the directory is encountered, a `NULL` pointer is returned and `errno` is unchanged.

If successful, `readdirplus_r()` returns `0`, and sets `result` to point to the same memory as entry. Otherwise, an error number is returned to indicate the error, and `result` is set to a `NULL` pointer.

#### ERRORS

The `readdirplus()` and `readdirplus_r()` functions may fail with the following errors:

##### E\_OVERFLOW

One of the values in the structure cannot be represented correctly.

##### EBADF

The `dirp` argument does not refer to an open directory stream.

##### ENOENT

The current position of the directory stream is invalid.

##### EIO

An error occurred while reading from the file system.

In addition, if the `readdir()` portion of the operation succeeds but the `lstat()` portion fails, the `d_stat_err` field may fail with the following errors:

##### EACCES

A component of the path prefix denies search permission.

##### EIO

An error occurred while reading from the file system.

##### ENAMETOOLONG

The length of a pathname exceeds `{PATH_MAX}`, or pathname component is longer than `{NAME_MAX}`.

ENOENT

A component of the pathname does not name an existing file.

E\_OVERFLOW

The file size in bytes or the number of blocks allocated to the file cannot be represented correctly in the `d_stat` structure.