Rocky Enterprise Linux 9.2 Manual Pages on command 'readdir.2'

$ man readdir.2

READDIR(2)   Linux Programmer's Manual   READDIR(2)

NAME

   readdir - read directory entry

SYNOPSIS

   int readdir(unsigned int fd, struct old_linux_dirent *dirp,

        unsigned int count);

   Note: There is no glibc wrapper for this system call; see NOTES.

DESCRIPTION

   This is not the function you are interested in. Look at readdir(3) for

   the POSIX conforming C library interface. This page documents the bare

   kernel system call interface, which is superseded by getdents(2).

   readdir() reads one old_linux_dirent structure from the directory re?

   ferred to by the file descriptor fd into the buffer pointed to by dirp.

   The argument count is ignored; at most one old_linux_dirent structure

   is read.

   The old_linux_dirent structure is declared (privately in Linux kernel

   file fs/readdir.c) as follows:

   struct old_linux_dirent {

```
        unsigned long d_ino;     /* inode number */

        unsigned long d_offset;  /* offset to this old_linux_dirent */

        unsigned short d_namlen; /* length of this d_name */

        char  d_name[1];         /* filename (null-terminated) */

    }
```

d_ino is an inode number. d_offset is the distance from the start of

the directory to this old_linux_dirent. d_reclen is the size of

d_name, not counting the terminating null byte ('\0'). d_name is a

null-terminated filename.

RETURN VALUE

On success, 1 is returned. On end of directory, 0 is returned. On er‐

ror, -1 is returned, and errno is set appropriately.

ERRORS

EBADF Invalid file descriptor fd.

EFAULT Argument points outside the calling process's address space.

EINVAL Result buffer is too small.

ENOENT No such directory.

ENOTDIR

    File descriptor does not refer to a directory.

CONFORMING TO

This system call is Linux-specific.

NOTES

Glibc does not provide a wrapper for this system call; call it using

syscall(2). You will need to define the old_linux_dirent structure

yourself. However, probably you should use readdir(3) instead.

This system call does not exist on x86-64.

SEE ALSO

getdents(2), readdir(3)

COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A

description of the project, information about reporting bugs, and the

latest   version   of   this page, can   be   found   at

https://www.kernel.org/doc/man-pages/.