



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'dirfd.3p' command

\$ man dirfd.3p

DIRFD(3P) POSIX Programmer's Manual DIRFD(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

dirfd ? extract the file descriptor used by a DIR stream

SYNOPSIS

```
#include <dirent.h>

int dirfd(DIR *dirp);
```

DESCRIPTION

The `dirfd()` function shall return a file descriptor referring to the same directory as the `dirp` argument. This file descriptor shall be closed by a call to `closedir()`. If any attempt is made to close the file descriptor, or to modify the state of the associated description, other than by means of `closedir()`, `readdir()`, `readdir_r()`, `rewinddir()`, or `seekdir()`, the behavior is undefined.

RETURN VALUE

Upon successful completion, the `dirfd()` function shall return an integer which contains a file descriptor for the stream pointed to by `dirp`. Otherwise, it shall return `-1` and shall set `errno` to indicate the error.

ERRORS

The `dirfd()` function may fail if:

`EINVAL` The `dirp` argument does not refer to a valid directory stream.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The `dirfd()` function is intended to be a mechanism by which an application may obtain a file descriptor to use for the `fchdir()` function.

RATIONALE

This interface was introduced because the Base Definitions volume of POSIX.1?2017 does not make public the `DIR` data structure. Applications tend to use the `fchdir()` function on the file descriptor returned by this interface, and this has proven useful for security reasons; in particular, it is a better technique than others where directory names might change.

The description uses the term "a file descriptor" rather than "the file descriptor". The implication intended is that an implementation that does not use an fd for `opendir()` could still `open()` the directory to implement the `dirfd()` function. Such a descriptor must be closed later during a call to `closedir()`.

If it is necessary to allocate an fd to be returned by `dirfd()`, it should be done at the time of a call to `opendir()`.

FUTURE DIRECTIONS

None.

SEE ALSO

`closedir()`, `fchdir()`, `fdopendir()`, `fileno()`, `open()`, `readdir()`

The Base Definitions volume of POSIX.1?2017, `<dirent.h>`

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of

Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group

2017

DIRFD(3P)