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 'dup.3p' command

$ man dup.3p

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

dup, dup2 ? duplicate an open file descriptor

SYNOPSIS

```
#include <unistd.h>
int dup(int fildes);
int dup2(int fildes, int fildes2);
```

DESCRIPTION

The dup() function provides an alternative interface to the service provided by fcntl() using the F_DUPFD command. The call dup(fildes) shall be equivalent to:

```
fcntl(fildes, F_DUPFD, 0);
```

The dup2() function shall cause the file descriptor fildes2 to refer to the same open file description as the file descriptor fildes and to share any locks, and shall return fildes2. If fildes2 is already a valid open file descriptor, it shall be closed first, unless fildes is equal to fildes2 in which case dup2() shall return fildes2 without closing it. If the close operation fails to close fildes2, dup2() shall

return -1 without changing the open file description to which fildes2 refers. If fildes is not a valid file descriptor, dup2() shall return -1 and shall not close fildes2. If fildes2 is less than 0 or greater than or equal to {OPEN_MAX}, dup2() shall return -1 with errno set to [EBADF].

Upon successful completion, if fildes is not equal to fildes2, the FD_CLOEXEC flag associated with fildes2 shall be cleared. If fildes is equal to fildes2, the FD_CLOEXEC flag associated with fildes2 shall not be changed.

If fildes refers to a typed memory object, the result of the dup2() function is unspecified.

RETURN VALUE

Upon successful completion a non-negative integer, namely the file de? scriptor, shall be returned; otherwise, -1 shall be returned and errno set to indicate the error.

ERRORS

The dup() function shall fail if:

EBADF The fildes argument is not a valid open file descriptor.

EMFILE All file descriptors available to the process are currently open.

The dup2() function shall fail if:

EBADF The fildes argument is not a valid open file descriptor or the argument fildes2 is negative or greater than or equal to {OPEN_MAX}.

EINTR The dup2() function was interrupted by a signal.

The dup2() function may fail if:

EIO An I/O error occurred while attempting to close fildes2.

The following sections are informative.

EXAMPLES

Redirecting Standard Output to a File S

The following example closes standard output for the current processes, re-assigns standard output to go to the file referenced by pfd, and closes the original file descriptor to clean up.

```
#include <unistd.h>
...
int pfd;
...
close(1);
dup(pfd);
close(pfd);
...
```

Redirecting Error Messages

The following example redirects messages from stderr to stdout.

```
#include <unistd.h>
...
dup2(1, 2);
...
```

APPLICATION USAGE

Implementations may use file descriptors that must be inherited into child processes for the child process to remain conforming, such as for message catalog or tracing purposes. Therefore, an application that calls dup2() with an arbitrary integer for fildes2 risks non-conforming behavior, and dup2() can only portably be used to overwrite file de? scriptor values that the application has obtained through explicit ac? tions, or for the three file descriptors corresponding to the standard file streams. In order to avoid a race condition of leaking an unin? tended file descriptor into a child process, an application should con? sider opening all file descriptors with the FD_CLOEXEC bit set unless the file descriptor is intended to be inherited across exec.

RATIONALE

The dup() function is redundant. Its services are also provided by the fcntl() function. It has been included in this volume of POSIX.1?2017 primarily for historical reasons, since many existing applications use it. On the other hand, the dup2() function provides unique services, as no other interface is able to atomically replace an existing file de? scriptor.

The dup2() function is not marked obsolescent because it presents a type-safe version of functionality provided in a type-unsafe version by fcntl(). It is used in the POSIX Ada binding.

The dup2() function is not intended for use in critical regions as a synchronization mechanism.

In the description of [EBADF], the case of fildes being out of range is covered by the given case of fildes not being valid. The descriptions for fildes and fildes2 are different because the only kind of invalid? ity that is relevant for fildes2 is whether it is out of range; that is, it does not matter whether fildes2 refers to an open file when the dup2() call is made.

FUTURE DIRECTIONS

None.

SEE ALSO

close(), fcntl(), open()

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