NAME

dup, dup2, dup3 − duplicate a file descriptor

SYNOPSIS

#include <unistd.h>

int dup(int *oldfd***);**
int dup2(int *oldfd***, int** *newfd***);**

#define _GNU_SOURCE
#include <unistd.h>

int dup3(int *oldfd***, int** *newfd***, int** *flags***);**

DESCRIPTION

These system calls create a copy of the file descriptor *oldfd*.

dup() uses the lowest-numbered unused descriptor for the new descriptor.

dup2() makes *newfd* be the copy of *oldfd*, closing *newfd* first if necessary, but note the following:

- If *oldfd* is not a valid file descriptor, then the call fails, and *newfd* is not closed.
- If *oldfd* is a valid file descriptor, and *newfd* has the same value as *oldfd*, then **dup2**() does nothing, and returns *newfd*.

After a successful return from one of these system calls, the old and new file descriptors may be used interchangeably. They refer to the same open file description (see **open**(2)) and thus share file offset and file status flags; for example, if the file offset is modified by using **lseek**(2) on one of the descriptors, the offset is also changed for the other.

The two descriptors do not share file descriptor flags (the close-on-exec flag). The close-on-exec flag (**FD_CLOEXEC**; see **fcntl**(2)) for the duplicate descriptor is off.

dup3() is the same as **dup2**(), except that:

- The caller can force the close-on-exec flag to be set for the new file descriptor by specifying **O_CLOEXEC** in *flags*. See the description of the same flag in **open**(2) for reasons why this may be useful.
- If *oldfd* equals *newfd*, then **dup3**() fails with the error **EINVAL**.

RETURN VALUE

On success, these system calls return the new descriptor. On error, −1 is returned, and *errno* is set appropriately.

ERRORS

EBADF
oldfd isn't an open file descriptor, or *newfd* is out of the allowed range for file descriptors.

EBUSY
(Linux only) This may be returned by **dup2**() or **dup3**() during a race condition with **open**(2) and **dup**().

EINTR
The **dup2**() or **dup3**() call was interrupted by a signal; see **signal**(7).

EINVAL
(**dup3**()) *flags* contain an invalid value. Or, *oldfd* was equal to *newfd*.

EMFILE
The process already has the maximum number of file descriptors open and tried to open a new one.

VERSIONS

dup3() was added to Linux in version 2.6.27; glibc support is available starting with version 2.9.

CONFORMING TO

dup(), **dup2**(): SVr4, 4.3BSD, POSIX.1-2001.

dup3() is Linux-specific.

NOTES

The error returned by **dup2**() is different from that returned by **fcntl(**..., **F_DUPFD**, ...**)** when *newfd* is out of range. On some systems **dup2**() also sometimes returns **EINVAL** like **F_DUPFD**.

If *newfd* was open, any errors that would have been reported at **close**(2) time are lost. A careful programmer will not use **dup2**() or **dup3**() without closing *newfd* first.

SEE ALSO

close(2), **fcntl**(2), **open**(2)

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.