NAME

dup, **dup2** - duplicate an existing file descriptor

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <unistd.h>

int
dup(*int oldd*);

int
dup2(*int oldd, int newd*);

DESCRIPTION

The **dup**() system call duplicates an existing object descriptor and returns its value to the calling process (*newd* = **dup**(*oldd*)). The argument *oldd* is a small non-negative integer index in the per-process descriptor table. The new descriptor returned by the call is the lowest numbered descriptor currently not in use by the process.

The object referenced by the descriptor does not distinguish between *oldd* and *newd* in any way. Thus if *newd* and *oldd* are duplicate references to an open file, read(2), write(2) and lseek(2) calls all move a single pointer into the file, and append mode, non-blocking I/O and asynchronous I/O options are shared between the references. If a separate pointer into the file is desired, a different object reference to the file must be obtained by issuing an additional open(2) system call. The close-on-exec flag on the new file descriptor is unset.

In **dup2**(), the value of the new descriptor *newd* is specified. If this descriptor is already in use and *oldd* != *newd*, the descriptor is first deallocated as if the close(2) system call had been used. If *oldd* is not a valid descriptor, then *newd* is not closed. If *oldd* == *newd* and *oldd* is a valid descriptor, then **dup2**() is successful, and does nothing.

RETURN VALUES

These calls return the new file descriptor if successful; otherwise the value -1 is returned and the external variable *errno* is set to indicate the cause of the error.

ERRORS

The **dup**() system call fails if:

[EBADF] The *oldd* argument is not a valid active descriptor

[EMFILE] Too many descriptors are active.

The **dup2**() system call fails if:

[EBADF] The *oldd* argument is not a valid active descriptor or the *newd* argument is negative or exceeds the maximum allowable descriptor number

SEE ALSO

accept(2), close(2), fcntl(2), getdtablesize(2), open(2), pipe(2), socket(2), socketpair(2), dup3(3)

STANDARDS

The **dup**() and **dup2**() system calls are expected to conform to IEEE Std 1003.1-1990 ("POSIX.1").

HISTORY

The **dup**() function appeared in Version 3 AT&T UNIX. The **dup2**() function appeared in Version 7 AT&T UNIX.