

SETIO(f)

SETIO(f)

NAME

setio – set I/O mode of file

SYNOPSIS

(setio = 75.)
(file descriptor in r0)
sys setio; mode
setio(fildes, mode)

DESCRIPTION

Setio sets the I/O mode of subsequent reads or writes to the file specified by the file descriptor *fildes* which is a word returned from a successful *open* on a file or *creat* of a new file. The possible modes which may be set for file I/O are:

- 0 - normal buffered I/O
- 01 - physical I/O directly to or from
the user's address space
- 02 - asynchronous I/O directly to or from the
user's address space

If asynchronous I/O to the user's address space is initiated, the I/O must eventually be waited for by a call to *statio*. It is possible to set the mode of all file descriptors used by a program in one call. When -1 is specified as the file descriptor, then the mode applies to all file descriptors. Thus, *setio(-1,1)* forces all file descriptors to use physical I/O, and a *setio(-1,2)* asynchronous I/O. A *setio(-1,0)* causes all file descriptors to revert to their previous modes.

In the case of physical I/O, if the I/O does not start on a device block boundary, i.e. a multiple of 256 words, normal system side-buffering is used. An I/O transfer may be broken up into a combination of physical I/O and buffered I/O by the system if this is possible. In the case of asynchronous I/O to or from the user's address space, the transfer must always start on a device block boundary but need not necessarily end on a block boundary.

SEE ALSO

open(II), *read*(II), *write*(II), *read*(c), *write*(c), *statio*(f).

DIAGNOSTICS

The error bit (c-bit) is set if the file descriptor is not that of an open file. From C, a -1 value is returned on an error.

BUGS

It is not wise in general, to write programs that use several file descriptors with different I/O modes to read and write data to a file.

It should be possible to turn off write behind on an individual file descriptor basis.