NAME

ioperm − set port input/output permissions

SYNOPSIS

#include <unistd.h> /* for libc5 */
#include <sys/io.h> /* for glibc */

int ioperm(unsigned long *from***, unsigned long** *num***, int** *turn_on***);**

DESCRIPTION

ioperm() sets the port access permission bits for the calling process for *num* bytes starting from port address *from* to the value *turn_on*. If *turn_on* is non-zero, the calling process must be privileged (**CAP_SYS_RAWIO**).

Only the first 0x3ff I/O ports can be specified in this manner. For more ports, the **iopl**(2) system call must be used.

Permissions are not inherited by the child created by **fork**(2). Permissions are preserved across **execve**(2); this is useful for giving port access permissions to non-privileged programs.

This call is mostly for the i386 architecture. On many other architectures it does not exist or will always return an error.

RETURN VALUE

On success, zero is returned. On error, −1 is returned, and *errno* is set appropriately.

ERRORS

EINVAL
Invalid values for *from* or *num*.

EIO (on PowerPC) This call is not supported.

ENOMEM
Out of memory.

EPERM
The calling process has insufficient privilege.

CONFORMING TO

ioperm() is Linux-specific and should not be used in programs intended to be portable.

NOTES

Libc5 treats it as a system call and has a prototype in <*unistd.h*>. Glibc1 does not have a prototype. Glibc2 has a prototype both in <*sys/io.h*> and in <*sys/perm.h*>. Avoid the latter, it is available on i386 only.

SEE ALSO

iopl(2), **capabilities**(7)

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/.