NAME

iopl − change I/O privilege level

LIBRARY

Standard C library (*libc*, *−lc*)

SYNOPSIS

#include <sys/io.h>

[[deprecated]] int iopl(int *level***);**

DESCRIPTION

iopl() changes the I/O privilege level of the calling thread, as specified by the two least significant bits in *level*.

The I/O privilege level for a normal thread is 0. Permissions are inherited from parents to children.

This call is deprecated, is significantly slower than **ioperm**(2), and is only provided for older X servers which require access to all 65536 I/O ports. It 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 to indicate the error.

ERRORS

EINVAL

level is greater than 3.

ENOSYS

This call is unimplemented.

EPERM

The calling thread has insufficient privilege to call **iopl**(); the **CAP_SYS_RAWIO** capability is required to raise the I/O privilege level above its current value.

STANDARDS

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

NOTES

glibc2 has a prototype both in *<sys/io.h>* and in *<sys/perm.h>*. Avoid the latter, it is available on i386 only.

Prior to Linux 5.5 **iopl**() allowed the thread to disable interrupts while running at a higher I/O privilege level. This will probably crash the system, and is not recommended.

Prior to Linux 3.7, on some architectures (such as i386), permissions *were* inherited by the child produced by **fork**(2) and were preserved across **execve**(2). This behavior was inadvertently changed in Linux 3.7, and won't be reinstated.

SEE ALSO

ioperm(2), **outb**(2), **capabilities**(7)