NAME

iopl − change I/O privilege level

SYNOPSIS

#include <sys/io.h>

int iopl(int *level***);**

DESCRIPTION

iopl() changes the I/O privilege level of the calling process, as specified in *level*.

This call is necessary to allow 8514-compatible X servers to run under Linux. Since these X servers require access to all 65536 I/O ports, the **ioperm**(2) call is not sufficient.

In addition to granting unrestricted I/O port access, running at a higher I/O privilege level also allows the process to disable interrupts. This will probably crash the system, and is not recommended.

Permissions are inherited by **fork**(2) and **execve**(2).

The I/O privilege level for a normal process is 0.

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
level is greater than 3.

ENOSYS
This call is unimplemented.

EPERM
The calling process has insufficient privilege to call **iopl**(); the **CAP_SYS_RAWIO** capability is required.

CONFORMING TO

iopl() is Linux-specific and should not be used in processes 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

ioperm(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/.