



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'iopl.2'

\$ man iopl.2

IOPL(2) Linux Programmer's Manual IOPL(2)

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 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 appropriately.

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.

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)`

COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.