

NAME

`ioperm` – set port input/output permissions

SYNOPSIS

```
#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 thread for `num` bits starting from port address `from`. If `turn_on` is nonzero, then permission for the specified bits is enabled; otherwise it is disabled. If `turn_on` is nonzero, the calling thread must be privileged (**CAP_SYS_RAWIO**).

Before Linux 2.6.8, only the first 0x3ff I/O ports could be specified in this manner. For more ports, the `iopl(2)` system call had to be used (with a `level` argument of 3). Since Linux 2.6.8, 65,536 I/O ports can be specified.

Permissions are inherited by the child created by `fork(2)` (but see NOTES). Permissions are preserved across `execve(2)`; this is useful for giving port access permissions to unprivileged 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 thread has insufficient privilege.

CONFORMING TO

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

NOTES

The `/proc/ioproports` file shows the I/O ports that are currently allocated on the system.

Before Linux 2.4, permissions were not inherited by a child created by `fork(2)`.

Glibc has an `ioperm()` prototype both in `<sys/io.h>` and in `<sys/perm.h>`. Avoid the latter, it is available on i386 only.

SEE ALSO

`iopl(2)`, `outb(2)`, `capabilities(7)`

COLOPHON

This page is part of release 4.16 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/>.