NAME

getrandom - get random data

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/random.h>

ssize_t
getrandom(*void *buf*, *size_t buflen*, *unsigned int flags*);

DESCRIPTION

getrandom() fills *buf* with up to *buflen* bytes of random data.

The *flags* argument may include zero or more of the following:

‘GRND_NONBLOCK’ Return EAGAIN instead of blocking, if the random(4) device has not yet been seeded. By default, **getrandom**() will block until the device is seeded.

‘GRND_RANDOM’ This flag does nothing on FreeBSD. */dev/random* and */dev/urandom* are identical.

‘GRND_INSECURE’ This flag is treated as an alternative name for GRND_NONBLOCK. It is provided solely for API compatibility with Linux.

If the random(4) device has been seeded, reads of up to 256 bytes will always return as many bytes as requested and will not be interrupted by signals.

RETURN VALUES

Upon successful completion, the number of bytes which were actually read is returned. For requests larger than 256 bytes, this can be fewer bytes than were requested. Otherwise, -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **getrandom**() operation returns the following errors:

[EAGAIN] The ‘GRND_NONBLOCK’ (or ‘GRND_INSECURE’) flag was set and the random(4) device was not yet seeded.

[EFAULT] The *buf* parameter points to an invalid address.

[EINTR] The sleep was interrupted by a signal.

[EINVAL] An invalid *flags* was specified.

[EINVAL] The requested *buflen* was larger than IOSIZE_MAX.

SEE ALSO

arc4random(3), getentropy(3), random(4)

STANDARDS

getrandom() is non-standard. It is present in Linux.

HISTORY

The **getrandom**() system call first appeared in FreeBSD 12.0.

CAVEATS

Unlike Linux, the GRND_INSECURE flag on FreeBSD does not produce any output before the random(4) device is seeded.